Ontwerp en optimalisatie van optische grids en clouds

Design and Optimization of Optical Grids and Clouds

Jens Buysse

Promotoren: prof. dr. ir. C. Develder, prof. dr. ir. B. Dhoedt
Proefschrift ingediend tot het behalen van de graad van
Doctor in de Ingenieurswetenschappen: Computerwetenschappen

Vakgroep Informatietechnologie
Voorzitter: prof. dr. ir. D. De Zutter
Faculteit Ingenieurswetenschappen en Architectuur
Academiejaar 2012 - 2013

UNIVERSITEIT
GENT

ISBN 978-90-8578-591-0
NUR 986
Wettelijk depot: D/2013/10.500/24

Universiteit Gent
Faculteit Ingenieurswetenschappen en Architectuur
Vakgroep Informatietechnologie

Promotoren: prof. dr. ir. Chris Develder
prof. dr. ir. Bart Dhoedt

Universiteit Gent
Faculteit Ingenieurswetenschappen en Architectuur

Vakgroep Informatietechnologie
Gaston Crommenlaan 8 bus 201, B-9050 Gent, België

Tel.: +32-9-331.49.00
Fax.: +32-9-331.48.99

Dit werk kwam tot stand in het kader van een specialisatiebeurs van het IWT-Vlaanderen (Instituut voor de aanmoediging van Innovatie door Wetenschap en Technologie in Vlaanderen).

Proefschrift tot het behalen van de graad van Doctor in de Ingenieurswetenschappen: Computerwetenschappen
Academiejaar 2012-2013

Weten waar Abraham zijn mosterd haalt

Op het moment van schrijven, zit ik aan een klein tafeltje te mijmeren aan een raam met zicht op de op dit moment nog besneeuwde bergtoppen van Alpe d'Huez; *Lady d'Arbanville* van Cat Stevens speelt op de achtergrond. Over twee weken geef ik de presentatie op de publieke verdediging en dat doet me nadenken: hoe ben ik tot dit punt geraakt? Herinneringen aan leuke maar ook minder leuke momenten doen de revue: hoe ik trots na een vergadering in Barcelona belde naar Emma om te zeggen dat ik met mijn voeten in de Middellandse zee stond, maar ook de nacht waar ik verwoed en uitgeteld resultaten van de cluster haalde om te beseffen dat er nog steeds een foutje in de programmatuur zat. Nee, niet alles verliep altijd van een leien dakje, maar hoe meer ik erover nadenk, hoe meer ik besef dat ik me een paar vuistregels voor ogen gehouden heb die me naar dit punt geleid hebben. Deze regels wil ik jullie niet onthouden beste familie, vrienden, collega's en eventuele toekomstige doctoraatsstudenten die dit boek ter hand genomen hebben als inspiratie voor hun eigen werk.

Quitters never win, winners never quit. But those who never win and never quit are idiots.

Deze slagzin komt van David Brent, het fictieve personage uit "The Office". Hoewel hij het voor totaal andere redenen zegt, heeft het wel betrekking tot het volbrengen van dit werk. Na het sollicitatiegesprek met Piet Demeester, Bart Dhoedt en Filip De Turck heb ik de kans gekregen om aan deze klus te beginnen, waarbij het behalen van de IWT beurs voor een stabiel financieel kader zorgde. Ik ben zowel IBCN als het IWT dankbaar voor deze kans, niettegenstaande er toch momenten waren dat ik met de gedachte speelde om er de brui aan te geven. Drie mensen hebben hier een cruciale rol gespeeld en hielpen die gedachten omzetten in een duidelijk motivatie.

De oudste herinnering die ik aan Marc De Leenheer heb is die waar ik hem onverwacht na de werkuren tegenkwam in café Dambert. Na wat bier gaf hij me eveneens een gouden raad: "Als je iets niet begrijpt, niet goed weet hoe aan te pakken of andere problemen hebt kom dan bij ofwel mij ofwel je promotor aankloppen - en blijf dit doen totdat je het antwoord hebt." Ik heb die raad steevast

opgevolgd. Een van die personen bij wie ik kon aankloppen, was Chris Develder, die me nauw opgevolgd heeft. Zonder jouw paperreviews en brainstormsessies had dit werk waarschijnlijk nooit tot stand kunnen komen. Bart Dhoedt maakte het plaatje compleet, met zijn opmerkelijke gave om na twee minuten volledig mee te zijn met het onderwerp en net dat gaatje op te vullen waar de student al menige uren zijn hoofd over gebroken heeft.

Mensen die je zowel door de modder als door regenplassen helpen

Als er iets is dat je als doctoraatstudent niet mag missen, zijn het mensen bij wie je terecht kan op moeilijke momenten, maar ook op gebeurtenissen waarbij vreugde gedeeld wordt. Bij mij is dat op eerste plaats Emma Matthys, die me onvoorwaardelijke gesteund heeft, ongeacht hoe chagrijnig ik ook thuis kwam na een dag van onsuccesvol zoeken naar die ene fout die zich maar om de 10000 events voordoet. Soms was de mogelijkheid om mijn hoofd op jouw schoot te mogen leggen meer dan genoeg om de volgende dag met volle moed terug op mijn fiets te kruipen. Zonder Emma, had ik hier waarschijnlijk niet gestaan. Dat geldt ook voor Johan Buysse, Elly Vervaart & Jannes Buysse, vader, moeder en broer, die bovenstaande taak overgenomen hebben wanneer Emma even niet te bespeuren was.

Markeer trouwens volgende naam: “Tim Raats”. Naast een van mijn beste vrienden is hij ook een partner in crime. Hij is op dit moment, naast enkele andere schrijfsels, zijn eigen doctoraatsproefschrift (communicatiewetenschappen) aan het voorbereiden. Samen Aspe bekijken, een film meepikken, Hommelbier drinken, pistes afglijden : he does it all! Samen met Emma ben ik hem veel verschuldigd en ik hoop van harte dat andere mensen ook dergelijke personen rondom zich mogen hebben.

Mensa Sana in Corpore sano

Ook al lijkt het of dat je je tijd toch beter besteedt aan het oplossen van het eigenlijke academische probleem, is het toch beter om even afstand te nemen en eens je lichaam goed op de proef te stellen. Bij mij was dit grotendeels in de vorm van mijn passie waterpolo. Wat begon als een maandagavondtraining, werd al vlug uitgebreid naar een donderdagavondtraining met af en toe zelfs (inter)nationale uitwedstrijden. Onze miskende trainer en tevens organisator van alles wat met waterpolo te maken heeft bij de UGent (en bij uitbreiding gans Gent), Willem Vercruysen, verdient daardoor ook een grote pluim. Hij zorgde ervoor dat ik dat balletje over en weer kon gooien met “ons Timmeke” waarna af en toe een belletje rinkelde en het oorspronkelijke vraagstuk de volgende ochtend eens anders ging aanpakken.

Nous ne sommes jamais “sans famille”

Als groentje ben ik bureau 2.22 binnen gegaan, als vader overste verlaat ik het. Veel mensen hebben er hun stekje gehad: Masha, Pieter & Peter (voor één of andere reden noem je die altijd samen) Jos, Johannes, de Dieters, Matthias, Heiko, Bart, Kevin en nog vele anderen. Hoewel mijn scriptieonderwerp vaak een ver van mijn bed show was voor hen, kon ik toch regelmatig bouwen op hun expertise, zoals hieronder waar we samen een C++ raadsel ontcijferen.

Op de derde verdieping, in een donker hoekje met harde Slayer basklanken op de achtergrond, heb je kans om “Slype” tegen het lijf te lopen. Tijdens mijn eerste jaar werken had ik zoals vele mensen redelijk wat schrik van deze kolos, maar die schrik veranderde met de jaren in een stevige vriendschap die we samen met Emma gevierd hebben op 12 meter diepte door eens van ademautomaat te wisselen.

Frameworks

Het leuke aan dit doctoraatsonderzoek is dat het grotendeels uitgevoerd is in het kader van het Europese GEYSERS project. Hiervoor werd ik vaak op missie gezonden naar projectmeetings, waardoor ik toch al mooie delen van Europa heb mogen bezichtigen en in contact gekomen ben met mensen die op hetzelfde onderwerp werken. Anna Tzanakaki en Kostas Georgakilas zijn er twee van. Ik heb de eer gehad om twee mooie en warme maanden in Athene bij hen op AIT te mogen werken, wat zowel voor een wereldlijke als een professionele verruiming gezorgd heeft, alsook hoofdstuk 5 in deze thesis.

Ter leering ende vermaak

Doctoreren betekent artikels schrijven en op conferenties gaan. Ik heb daarbij veelvuldig het nuttige aan het aangename gekoppeld: kitesurfen in Maui, late night comedy in New York, bezoek van de Gouden Tempel in Kyoto, Cirque du Soleil

bezichtigen in Montreal . . . en eenmaal werd het aangename teruggekoppeld aan het nuttige: tijdens een rondleiding in Washington ben ik aan de praat geraakt met Brigitte Jaumard en uit dat gesprek is uiteindelijk een groot gedeelte van hoofdstuk 4 gekomen.

En tot slot, enkele losse raadgevingen

- Probeer zo veel mogelijk acroniemen vanbuiten te kennen, dat maakt altijd indruk.
- Zoek een plaatsje buiten je werkomgeving waar je in de zomer even gezellig kan vertoeven (ik heb veel "patatten geplant" met David Plets)
- Zorg dat je de sleutel van je bureau altijd bij je hebt, zodat je niet buitengesloten wordt.
- Bij het ophalen van examenfiles, zorg ervoor dat je een goed werkende USB stick hebt.
- . . .

Ter afsluiting, probeer niemand te vergeten bij het schrijven van je dankwoord!

Gent, April 2013
Jens Buysse

Table of Contents

List of Figures

List of Tables

List of Acronyms

0-9

3R	Reshaping, Reamplification and Retiming

A

AA	Assisted Anycast
ASON	Automatically Switched Optical Network

B

BER	Bit Error Rate
BRPC	Backward Recursive Path Computation

C

CFD	Computational Fluid Dynamics
CG	Column Generation
CPU	Central Processing Unit
CRAH	Computer Room Air Handlers
CSP	Classical Shared Path protection
CSP-A	Classical Shared Path Protection under Anycast
CWDM	Coarse WDM

D

DEMUX	Demultiplexer

DFA	Doped Fiber Amplifier
DVS	Dynamic Voltage Scaling
DWDM	Dense WDM

E

EDFA	Erbium-Doped Fiber Amplifier
EE	Energy Efficient
EGEE	Enabling Grids for E-sciencE
ERO	Explicit Route Object

F

FA	Full Anycast
FCFS	First Come First Served
FDL	Fiber Delay Line
FF	Flow Formulation
FLOPS	Floating-point Operations Per Second
FM	Full Mesh

G

GB	Gigabyte
GLUE	Grid Laboratory Uniform Environment
GMPLS	Generalized Multi-Label Protocol Switching
G^2MPLS	Grid GMPLS

H

HDTV	High-Definition Television
H-PCE	Hierarchical PCE

I

IaaS	Infrastructure-as-a-Service
ICT	Information and Communications Technology

ILP	Integer Linear Programming
IP	Internet Protocol
IT	Information Technology
IT-S	IT Site
IT-M	IT Manager

L

LAN	Local Area Network
LICL	Logical Infrastructure Layer
LHC	Large Hadron Collider
LSP	Label Switched Path
LSU	Label State Update

M

MB	Megabyte
MEMS	Mircoelectromechanical Systems
MILP	Mixed Integer Linear Programming
MPLS	Multiprotocol Label Switching
MUX	Multiplexer

N

NCP	Network Control Plane
NIPS	Network and IT Provisioning
NT	Notification Type
NV	Notification Value

O

OADM	Optical add-drop Multiplexer
OCCI	Open Cloud Computing Interface
OCS	Optical Circuit Switching
OEO	Optical-to-Electronic-to-Optical
OLT	Optical Line Terminal
ONU	Optical Network Unit
OSPF-TE	Open Shortest Path First - Traffic Engineering

OXC	Optical Cross Connect

P

P2P	Peer-to-Peer
P2P-PCE	Peer-to-Peer PCE
PaaS	Platform-as-a-Service
PC	Personal Computer
PCC	Path Computation Client
PCE	Path Computation Element
PCEP	PCE Protocol
PCntf	PCEP Notify
PD-PCE	Per Domain PCE
PDU	Power Distribution Unit
PI	Physical Infrastructure
P-NNI	Private Network-to-Network Interface
PON	Passive Optical Network
PUE	Power Usage Effectiveness

Q

QoE	Quality of Experience
QoS	Quality of Service

R

REST	REpresentational State Transfer
RF	Route Formulation
RFC	Request For Comments
ROADM	Reconfigurable Optical Add-Drop Multiplexer
RSVP-TE	Resource Reservation Protocol - Traffic Engineering
RWA	Routing and Wavelenght Assignment
Rx	Receiver

S

SaaS	Software-as-a-Service
SLA	Servce Level Agreement

SMF	Single Mode Fiber
SP	Shared Path Protection
SPR	Shared Path Protection with Relocation
SPR-A	Shared Path Protection with Relocation under Anycast
SLA	Service Level Agreement

T

Tb	Terabit
TB	Terabyte
TCP	Transmission Control Protocol
TDM	Time Division Multiplexing
TE	Traffic Engineering
TED	Traffic Engineering Database
TLV	Type Length Value
Tx	Transmitter

U

UNI	User-to-Network-Interface
UNI-C	User-to-Network-Interface Client
UNI-N	User-to-Network-Interface Network
UPS	Uninterruptible Power Supply

V

VI	Virtual Infrastructure
VO	Virtual Observatory
VO	Virtual Organisation
VPN	Virtual Private Network
VSPT	Virtual Shortest Path Tree
VWP	Virtual Wavelength Path

W

WAN	Wide Area Network
WDM	Wavelength Division Multiplexing

WP	Wavelength Path
WSS	Wavelength Selective Switch
WWW	World Wide Web

X

XaaS	Anything-as-a-service where X is either, Software, Platform or Infrastructure

List of Definitions

1+1 protection	Protection strategy where a primary path is protected by a dedicated backup path.
1:1 protection	Protection strategy where a primary path is protected by a backup path. Backup paths can share resources (i.e., wavelengths) as long as their primary paths are link disjoint.
Anycast	A routing methodology where the ending node of a connection can be any of a set of possible destinations.
Channel	A communications path or the signal sent over that path. In WDM technology, a channel is assigned to a specific wavelength also termed lambda.
Cloud computing	Cloud computing provides easy access to a large pool of resources (hardware or software), which can be easily (re)configured to cope with the offered load on a pay-as-you-use basis. The properties of the offered resources are guaranteed by means of SLAs.
(D)WDM	(Dense) wavelength division multiplexing. The transmission of multiple signals over closely spaced wavelengths in the 1550-nm region on a single fiber.
Discrete event simulation	A discrete-event simulation is an approach based on the assumption that the operation of the system can be described as a discrete sequence of events in time. Each event occurs at a particular instant of time and marks a change of state of the system.
Fiber	The structure that guides light in a fiber optic system.
Grid Computing	Grid computing entails coordinated resource sharing and problem solving in dynamic, geographically dispersed, multi-institutional organizations using standard, general protocols and interfaces.
Lambda	A wavelength used to carry one or more data channels in a WDM or DWDM system. Also called wavelength.
Opaque optical network	Optical network that tries to avoid, but still includes optical/electrical/optical conversion.

Photonic	A term used to describe communications using photons, analogous to electronic for electronic communications.
Quality of Experience	Quality of Experience is a measure of the overall level of customer satisfaction, both objectively and subjectively.
Quality of Service	In networks, Quality of Service is the idea that transmission rates, error rates, and other characteristics can be measured, improved and to some extent guaranteed in advance.
Transparent optical network	Optical network where signals are never converted to the electrical domain between network ingress and egress.
Translucent optical network	Optical network where signals are either switched by the optical switch or by the electronic one.
Virtual Wavelength Path	A group of one or more channels between source and destination nodes. The term virtual indicates that the signal is transported on different physical wavelengths throughout the network.
Wavelength continuity constraint	A lightpath operating on the same wavelength across all fiber links, is said to obey the wavelength continuity constraint.
Wavelength conversion	An optical cross connect which is able to convert a wavelength on an incoming fiber to another wavelength on an outgoing fiber, is said to perform wavelength conversion.

Samenvatting
– Summary in Dutch –

Rond 1960 heeft John McCarthy de term “utility computing” geïntroduceerd: rekenkracht, dataopslag en alles wat met computers te maken heeft moet aangeboden kunnen worden als een dienst. Dit kan geïllustreerd worden aan de hand van een vergelijking met gas en water waar toegang verleend wordt door eenvoudigweg een kraantje open te draaien. Op een zelfde, eenvoudige manier zouden deze computationele bronnen op aanvraag moeten kunnen geleverd worden. Dit idee werd pas echt in de praktijk toegepast met de introductie van grid en cloud computing. In een dergelijke architectuur wordt een applicatie niet lokaal aangepakt (bijvoorbeeld op de PC van de gebruiker), maar wordt de verwerking ervan overgelaten aan de grid/cloud.

Deze applicaties kunnen onderverdeeld worden in drie categorieën: wetenschappelijke (o.a. deeltjesfysica), zakelijke- (o.a. bankensector) en consumentenapplicaties (o.a. online gaming). Hoewel deze categorieën verschillende doelstellingen beogen, hebben ze wel allemaal één eigenschap gemeen: ze vereisen grote dataoverdrachten op een betrouwbare, efficiënte en snelle manier. Daarom is het noodzakelijk dat het netwerk, dat een grid of cloud ondersteunt, deze verwachtingen kan inlossen. Optische netwerken, gebruik makend van Wavelength Division Multiplexing (WDM), hebben een lage latentietijd en bieden hogebandbreedteverbindingen aan. Deze optische netwerken zijn dan ook ideale kandidaten om grids en clouds van connectiviteit te voorzien. Dit heeft geleid heeft tot zogenaamde “optische grids” en “optische clouds”.

Tijdens de ontwikkeling van deze grids en clouds werd verondersteld dat het netwerk altijd voorhanden zou zijn. Men verwachtte niet dat netwerkcapaciteit (bandbreedte) ontoereikend kon zijn om een bepaalde dataoverdracht te realiseren. De onderzoeksvelden van grid/cloud computing enerzijds en optische netwerken anderzijds hebben zich daarom in het verleden ook grotendeels onafhankelijk van elkaar ontwikkeld. Gezien de werkbelasting van een grid/cloud zulke hoge eisen stelt betreffende connectiviteit, moet er aandacht besteed worden aan de integratie en optimalisatie van netwerk- en computationele bronnen.

Dit is ook de focus van dit werk: hoe kunnen we een infrastructuur die bestaat uit een optisch netwerk en IT hardware (optische grid/cloud) optimaliseren door zowel het netwerk als de IT hardware tegelijkertijd te beschouwen in de optimalisatiestap? Deze integratie hebben we toegepast bij oplossing van drie respectievelijke vraagstukken. Een eerste bekommernis bij het onwikkelen van grid/clouds,

is het omgaan met fouten/falingen die zich nu eenmaal onvermijdelijk in een dusdanige context voordoen (bv. links die kapot gaan bij wegenwerken). Hiervoor hebben we een aanpassing van een bestaand protectiemechanisme ontwikkeld, namelijk gedeelde padprotectie met relocatie. Hierbij mogen een primair- en een secundair pad (dat gebruikt wordt bij een netwerkfout) eindigen in een verschillende locatie, in tegenstelling tot de gebruikelijke gedeelde padprotectie onwikkeld voor punt-naar-punt gegevensoverdrachten. Dit is mogelijk omdat het anycast principe geldt in clouds: een gebruiker bekommert zich niet om de locatie waar zijn applicatie uitgevoerd wordt, zolang het gewenste resultaat maar binnen een beperkt tijdsbestek ontvangen wordt. Ten tweede onderzochten we het minimaliseren van energiegebruik in grids en clouds. We bereiken dit door enerzijds hardware die niet gebruikt wordt uit te schakelen en anderzijds, nogmaals het anycast principe toe te passen. Het laatste luik omvat een beschrijving van een geïntegreerde controle- en managementarchitectuur gebaseerd op een hiërarchisch "padberekeningselementsysteem", dat gecombineerde routering- en planningsberekingen kan maken.

Om de voorgestelde algoritmes en architectuur te kunnen evalueren, hebben we gebruik gemaakt van discreet-gebeurtenissensimulaties. Hiervoor hebben we een eigen simulatieomgeving gemaakt, gebaseerd op Omnet+. Deze simulatieomgeving hebben we vervolgens gebruikt om gemiddeld tot grote probleeminstanties te simuleren. In dit werk beschrijven we de architectuur en implementatie van deze omgeving, die aangewend is in de studies in de daaropvolgende hoofdstukken. We bespreken nu kort de drie vraagstukken hierboven opgesomd.

Om ervoor te zorgen dat grids en clouds niet volledig sneuvelen bij het falen van een enkele linkfout, hebben we een bestaand beschermingsmechanisme, namelijk gedeelde padbescherming, uitgebreid naar gedeelde padbescherming met relocatie. Gebruikmakend van het anycast principe (het maakt gebruikers niet uit waar hun taak terecht komt) laten we toe dat een primair pad en een steunpad eindigen in een verschillend datacenter, in tegenstelling tot het traditionele gedeelde padbescherming waar het eindpunt van het steunpad hetzelfde moet zijn als dat van het primaire pad. Om zulke configuraties te berekenen, hebben we drie methoden bedacht. De eerste is gebaseerd op een ILP-formulering die ondanks het maken van optimale berekeningen er niet in slaagt dit te doen voor middel- tot grote netwerken. Daarom hebben we ook twee schaalbare heuristieken ontwikkeld, die sub-optimale oplossingen berekenen, maar dit wel doen in een aanvaardbaar tijdsbestek. Als laatste hebben we een techniek toegepast die gebaseerd is op kolomgeneratie. Deze techniek levert betere oplossingen dan de heuristieken, in een nog steeds aanvaarbare tijd. We hebben onderzocht welke besparingen het door ons voorgestelde beschermingssysteem kan bieden en tonen aan in welke situaties de desbetreffende berekeningstechniek gebruikt kan worden.

In een tweede luik hebben we onderzocht hoe het energieverbruik van grids en clouds gereduceerd kan worden. Hiervoor hebben we een energie-efficiënt routeringalgoritme ontwikkeld. Deze energiereductie kan behaald worden op twee manieren: (i) door het uitschakelen van componenten wanneer die niet in gebruik zijn en (ii) door het uitbuiten van anycast om de meeste geschikte locatie van uitvoering te zoeken. Hiertoe hebben we eerst een energiemodel ontwikkeld, dat

zowel IT en netwerk energie modelleert. Dit model hebben we dan toegepast in ons online algoritme. We tonen aan dat minimaal energieverbruik niet kan bereikt worden aan de hand van algoritmes die zich hoofdzakelijk op het gebruik van IT of netwerkhardware focussen, maar wel door een geïntegreerd algoritme (Full Anycast) dat een zorgvuldige overweging maakt met betrekking tot de balans tussen netwerk - en IT energieparameters. Daarenboven demonstreren we dat de doeltreffendheid van het geïntegreerd algoritme niet kan benaderd worden door meer gebruikelijke technieken die eerst plannen en daarna een routeringstap uitvoeren.

Om cloud- en gridvoorzieningen aan te bieden, moet een provider aanzienlijk wat investeren om de platformen die de heterogene netwerk- en IT componenten beheren, te integreren. Daarom is het noodzakelijk dat een controlesysteem uitgedokterd wordt, dat het beheer op zich kan nemen. Dit is de laatste contributie van dit werk: een voorstel voor een controlesysteem, NCP+ genaamd, gebaseerd op Generalized Multi-Protocol Label Switching (GMPLS) met een hiërarchisch padberekeningselementarchitectuur, die het mogelijk maakt om routering- en planningsbeslissingen te maken op een snelle manier. We beschrijven (i) de architectuur van deze NCP+, (ii) de protocoluitbreidingen om IT-informatie te verspreiden, (iii) twee abstractiemethodieken die IT-informatie ook in rekening brengt, inherent aan de hierarchisch padberekeningselementarchitectuur en (iv) voorstellen voor routering- en planningsalgoritmes die gebruikt kunnen worden in de abstractiemethodieken. We tonen aan dat een volledig vermaasde voorstelling, Full Mesh genaamd, de beste voorstellingsmethodiek is. Hoewel deze minder schaalbaar is met betrekking tot de berekeningstijd, slaagt Full Mesh er toch in om efficiënte paden te berekenen gebruik makende van de voorgestelde routering- en planningsalgoritmen.

Summary

In 1960, John McCarthy introduced the term “utility computing”. The main idea was that computing resources could be sold through the utility business model, just like water and electricity is offered today. This concept really started to take shape with the introduction of grid and cloud computing paradigms, where users offload their work from their local host to a distributed computing system, actually pushing computational functionality into the network.

This workload has been divided into three categories: (i) so called e-Science applications (e.g. particle physics), (ii) business applications (e.g. financial institutions) and (iii) consumer applications (online gaming). They all have one thing in common: they require a predictable service and high capacity, on-demand data delivery. Consequently, a network supporting a grid or cloud network should be able to carry large data transfers in a fast and reliable way. Given their high data rates and low latency, optical networks based on wavelength division multiplexing (WDM) technology are ideally suited to support these grids and clouds, thus giving rise to so-called “optical grids” and “optical clouds”.

During the development of these grids and clouds, the network was supposed to be always available and consequently, research in the IT and the network worlds evolved largely independently from each other. However the workload offered to grid and cloud infrastructures is composed of high-performance and high-capacity network based applications, which the best effort Internet intrinsically cannot support anymore. Therefore, is has become a necessity to optimize the network and IT resources together and as such, jointly optimizing the network and IT infrastructure.

Network and IT convergence (and optimization) are principal to this work: how can we optimize the complete infrastructure by considering network and IT resources equally valuable, considering them both in the optimization stage? We have applied this integration paradigm in three aspects. A first major concern in deploying optical grids is resilience: ensuring service continuity under failure conditions is of utmost importance. For this we have proposed a protection strategy, where end points of a primary and backup network path can be different, as opposed to classical strategies which impose them to be the same. A second issue we study, is the reduction of energy consumption in the context of grid/clouds, by proposing energy efficient routing and scheduling. Finally, we propose an enhanced network control plane based on a hierarchical path computation element architecture, which is able to make efficient routing and scheduling decisions for such a networked IT infrastructure.

We identified discrete-event simulation as the most adequate method of evaluation for most of our proposed algorithms. Therefore, we have built a simulation environment, based on Omnet+, which we have used to perform validation and testing on medium to large infrastructure instances (consisting of tens of entities). We report on the design and implementation of this simulation environment, and demonstrate its features throughout the different simulations studies which have been conducted in this work. We now will highlight each of the three aforementioned topics.

First, to ensure service continuity, we have optimized the classical shared path protection strategy for the optical grid/cloud scenario by exploiting the anycast routing principle typical of grid scenarios (denoted as shared path protection with relocation (SPR)). Anycast states that a user requiring a service only cares about timely and correct processing, but is indifferent to the location of the execution of the service. Hence, instead of reserving a backup path to the resource indicated by the scheduler under failure-free conditions, it could be better to relocate the requested service to another resource if this implies network resource savings. We propose an optimal ILP formulation yielding optimal results, as well two scalable but suboptimal heuristics, and a scalable, nearly optimal method based on column generation in order to compute these primary and backup paths. We identify the potential resource savings enabled by SPR and investigate which method to use in which case.

In a second study, we aim to facilitate the energy efficient operation of an integrated optical network and IT infrastructure. In this respect we propose an energy-efficient routing algorithm for IT service requests originating from specific source sites that need to be executed by suitable IT resources (e.g., data centers). We are able to reduce the overall energy consumption of such an infrastructure by employing two strategies: (i) switch off components when they are idle and (ii) exploit the anycast principle to handle IT requests to the most appropriate data center and compute routes in an energy-efficient way. To achieve this, we created an energy model including both the network and IT resources, which we then employed in an online heuristic. We conclude that there is no "universally best" option amongst IT-only or network-only energy optimization and that we consequently need to consider both resources jointly in an integrated approach. This approach (Full Anycast) includes a careful consideration of combined network and IT resource energy parameters. Moreover, the minimal energy consumption can only be reached using Full Anycast as it is not possible to reach the same optimum with a simple two-step (first planning, then routing) heuristic. (Although for high load scenarios scheduling to the closest IT site with shortest path routing effectively approximates this optimum).

In order to provide cloud services, a provider is required to make substantial investments in the integration of the variety of platforms operating over the heterogeneous resources. Thus there is a need for an automated and combined control mechanism for IT and network resource provisioning to ensure service continuity, efficient use of resources, service performance guarantees, scalability and manageability. Finally, we propose a control system called NCP+: a GMPLS control plane

with a Hierarchical Path Computation Element (PCE) architecture able to jointly make network routing and IT server provisioning decisions. We discuss (i) the architecture of the NCP+, (ii) the protocol extensions necessary to accommodate IT advertisements, (iii) two IT-aware aggregation mechanisms to be used in the hierarchical PCE approach and, (iv) routing and scheduling algorithms for those aggregation mechanisms. We demonstrate that Full Mesh aggregation, where the domain topology is represented by a full mesh graph, although being less scalable in terms of computation time, is able to provision more efficiently than a simpler Star strategy, using the proposed load balancing routing and scheduling policy.

1
Foreword

"O Deep Thought computer" he said, "the task we have designed you to perform is this. We want you to tell us...." he paused, "The Answer."
"The Answer?" said Deep Thought. "The Answer to what?"
"Life!" urged Fook.
"The Universe!" said Lunkwill.
"Everything!" they said in chorus.
Deep Thought paused for a moment's reflection.
"Tricky," he said finally.
"But can you do it?"
Again, a significant pause.
"Yes," said Deep Thought, "I can do it."
"There is an answer?" said Fook with breathless excitement.
"Yes," said Deep Thought. "Life, the Universe, and Everything. There is an answer. But, I'll have to think about it."
Fook glanced impatiently at his watch.
"How long? he said.
"Seven and a half million years, said Deep Thought.
Lunkwill and Fook blinked at each other.
"Seven and a half million years...!" they cried in chorus.
"Yes, declaimed Deep Thought, "I said I'd have to think about it, didnt I?"

– Douglas Adams, The Hitchhiker's Guide to the Galaxy

1.1 Distributed computing

The term distributed computing can be defined as follows [1].

> "In general, distributed computing is any computing methodology, which involves multiple interacting processes remote from each to solve a computation or information processing problem."

But before we go into more detail on how such cooperation between these computers can be established, we need to ask the question: "Why would we require these computers to interact with each other in the first place?" Scientists today have hit a wall: they are faced with complex problems which a single modern computer cannot solve anymore. One could be tempted to wait for faster processors and larger computer capacity, but this would only put the problem off for a short period of time. Consequently, there is a need for a scalable system able to tackle these complex problems and this is where distributed computing comes into play. Distributed computing takes a large problem, breaks it into smaller units, and allows many computing nodes (able to communicate with each other) to work on the problem in parallel.

Moreover, distributed computing is also attractive from a commercial point of view. If a company needs to deploy an application, it could opt to buy computer and network equipment and manage it at its own premises. This requires not only a large capital investment, but this infrastructure needs to be maintained and leads to an increase in power consumption for the company. However, a second choice could be to outsource this to a specialized company which rents out the required infrastructure and bills on a pay-as-you-use basis. The latter choice changes the game for the firm as its IT infrastructure can scale with the size of the company, while operating the IT systems more cheaply.

1.1.1 History of distributed computing in a nutshell

The concept of such utility computing, i.e., time-sharing of computer technology, was first introduced by John McCarthy in a speech given to celebrate MIT's centennial. The idea is that computing power, storage space or even applications could be sold through the utility business model (just like water or electricity). This concept became very popular and led to the idea that applications need to be pushed further into the network, leading to different distributed computing technologies. This work investigates methodologies and algorithms which can be used in this kind of utility computing, namely in grid and cloud computing environments. Before we explain these concepts, we need to go back in time.

It all started with the introduction of big mainframes in the 1940-60s, such as the ENIAC and Mark II (which had the first actual bug). In essence, they were

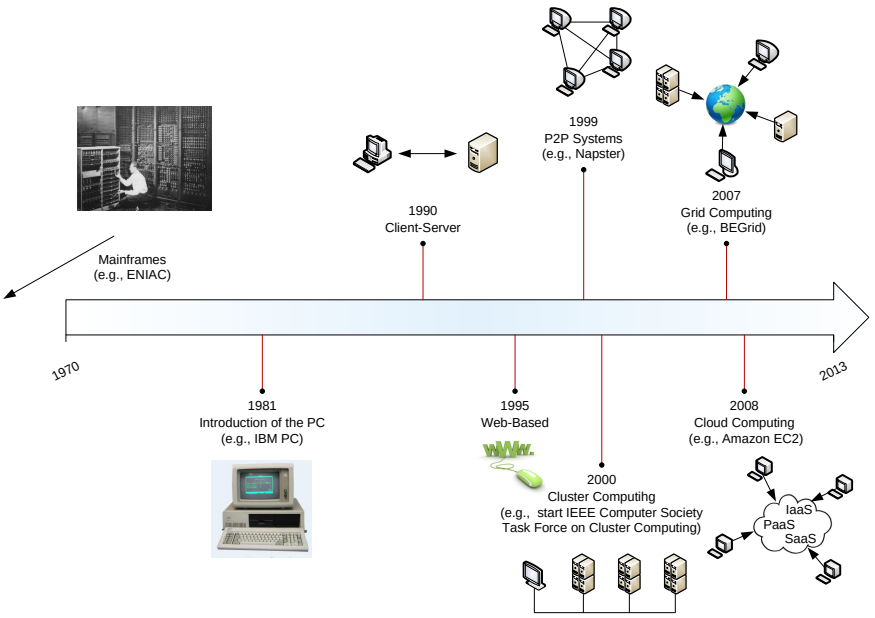


Figure 1.1: In order to cope with ever increasing requirements of applications, distributed computing paradigms have been devised, where these requirements are met by means of time-sharing of computing technology.

grotesque monsters, weighing over a couple of tons. Their selling point was that they provided the users with everything necessary to fulfill the task at hand. If there was any communication between mainframes, it amounted to transferring data by means of tape from one mainframe to another.

In the 1980s however, the personal computer (PC) saw the light of day. Initially the idea of the PC was to use it as a stand-alone device, but the introduction of the Internet Protocol Suite (TCP/IP) made it possible to interconnect PCs allowing them to communicate with each other. This gave rise to the Client-Server architecture, where a client machine asks for a server's content or service. An example of this is the Internet as we know it: a machine with a web browser asking another machine (the server) to get access to a certain website.

However, as more and more users were introduced to the Internet, there was a need to share files and computation power. This instigated the introduction of the Peer-to-Peer (P2P) architecture. A P2P system is composed of distributed desktop machines or servers sharing a portion of their processing power or storage resources, typically within a Wide Area Network (WAN). In such a system, each entity is a supplier and consumer of resources at the same time. One example of such a system is SETI@Home [2], which is a volunteer computing project looking for the answer whether there is extraterrestrial intelligence. Anybody who runs

the program, downloads radio-telescope data and starts analyzing it, exploiting the otherwise unused processor time.

Another distributed computing paradigm which popped up, was cluster computing. Clusters consist of a set of homogeneous servers interconnected by a Local Area Network (LAN). In most cases, the servers are owned by a single organization and are used as computational/storage resources for local users. In a sense, cluster computing resembles a supercomputer, which is a computer with a large number of processors interconnected by a high-speed bus. However, cluster computing has the advantage of scalability: it is easy for a cluster operator to upgrade the cluster by adding and/or replacing servers. An example of such a cluster is the HPC UGent cluster [3] (which has been extensively used throughout this PhD).

Building on the cluster computing concept, grid computing was introduced. Grid computing became recognized when Ian Foster and Carl Kesselman published "The Grid: Blueprint for a new computing infrastructure" [4]. A grid infrastructure comprises heterogeneous, geographically distributed computational resources from different administrative domains: clusters, servers, supercomputers, etc. Its main principle is that instead of buying extra resources to cope with peak loads, organizations can make use of the idle resources of other organizations, possibly subject to payment. Important examples of grid computing projects are the Open Science Grid [5], TerraGrid [6] and the EGEE [7] project. The latter, is mainly used to process the data from the Large Hadron Collider (LHC) built at CERN to study the fundamental properties of subatomic particles and forces.

Finally, at the end of the line, we find cloud computing which has received a lot of growing interest, not only from the academic but also from the business world. In cloud computing, the idea of McCarthy is applied to its full extent: users can request anything as a service, amounting to the *X-as-a-Service paradigm*. The X is either a complete IT infrastructure (Infrastructure-as-a-Service - IaaS), a software development platform (Platform-as-a-Service - PaaS) or applications and software (Software-as-a-Service - SaaS). Cloud computing suits the need of a company to dynamically scale: when the company grows or gets smaller, the requested amount of resources can scale with it. Organizations are not obliged to own their own IT infrastructure anymore and are able to share computing and capacity on an as-needed basis. This reduces an organization's capital and opertional expense substantially, with examples of 50% [8] and 80% [9] of reduction. A well-known SaaS provider is Salesforce.com [10] which is a company offering customer relationship management software and was one of the first companies to provide SaaS. A nice example of PaaS is Amazon Elastic Compute Cloud (Amazon EC2) [11] which provides resizeable computing capacity to enable web-scale computing. The same company offers IaaS, namely Amazon Web Services [12]. It offers a complete set of infrastructure and application services that enable you to run virtually everything in the cloud: from enterprise applications and big data projects to social games and

mobile apps.

1.1.2 Treating the network as a valuable resource

A key role in these distributed computing architectures, is played by the networks interconnecting them. Whether it is a LAN (Ethernet, WiFi) or a WAN (ATM), without communication, distributed processing cannot be performed. In the early distributed computing developments, the network (i.e., bandwidth) was not considered as a limiting factor which led to innovations in the telecommunication and the computing world to be isolated from each other. This seperation led to serious problems for the architects and designers of distributed systems. Illustrative are the so called "Eight fallacies of distributed computing" [13] which are not coincidentally network related:

1. The network is reliable.
2. Latency is zero.
3. Bandwidth is infinite.
4. The network is secure.
5. Topology doesn't change.
6. There is one administrator.
7. Transport cost is zero.
8. The network is homogeneous.

Distributed computing applications today have both strong computational/storage and bandwidth requirements, which cannot run effectively if these issues are not addressed. Consider HDTV, multiplayer video gaming and video conferencing: they require low latency and high bandwidth connections making the "always available" feature of the supporting network not that obvious anymore.

This results in the observation that network resources should be treated as a valuable service instead of taking them for granted. This convergence has received a lot of attention, especially in the context of grid and cloud computing. These systems view their resources, whether computing, storage or network resources as part of a shareable, common resource pool which is orchestrated by the same management system. As a result, today's telecom providers are faced with an increasing need for providing dynamic, high capacity and high performance connectivity, tightly bundled with IT resources.

The optical network, is undeniably a perfect candidate to support distributed architectures considering its low latency and high bandwidth properties. A grid or

cloud infrastructure, supported by an optical network is called an optical grid or optical cloud and these infrastructure are the core topic of this work. However, the seamless integration of network and computing resources poses a lot of interesting challenges which need to be addressed in order to support resiliency, meet Service Level Agreements (SLA) and run the infrastructure efficiently. The aim of this PhD is providing solutions to address these challenges.

1.2 Challenges in optical grid/cloud computing

In this section, we dicuss the challenges in grid and cloud computing that have been addressed in this PhD research. Note that this is not a restrictive list, but merely a subset of technical challenges selected from [14, 15].

- **Unified management of network and IT resources** - Traditionally, a service provider aiming to offer grid/cloud services is required to substantially invest in the integration of the variety of platforms operating over the heterogeneous resources within his management system. Thus there is a need for an automated and combined control mechanism for IT and network resources to ensure service continuity, efficient use of resources, service performance guarantees, scalability and manageability. These goals can be achieved either by reusing and combining existing separate IT and network management systems, or by developing new joint platforms. However, the former solution does not avoid human intervention (as configurations need to be made manually), and the efficiency of the whole system is bound by the limits of the separate components [16]. Instead, deploying an integrated control plane would enable both scalability and efficient operation over the network and IT infrastructure.

- **Multi-domain issues** - The Internet as we know it, is not one network but rahter a "network of networks". Consequently, telecom operators have several independent domains based on diverse technology standards and protocols, making inter-domain service provisioning extremely difficult. Therefore it is critical that we come up with a standardized interworking across diverse networks, allowing end-to-end service provisioning while achieving a cost-efficient infrastructure operation.

- **Scalability** - Scalability refers to the property of a system to handle a growing demand or work flow without drastic reduction in service quality. On the one hand, a grid/cloud infrastructure should be able to support multiple users providing them with the best possible response time, while still maximizing the utilization of the resources. On the other hand, grid/cloud computing should provide a solution to support business growth, as well as

addressing the needs for small clients providing them with the same Quality of Experience (QoE).

- **Seamless and coordinated provisioning of network and IT resources** - We have already mentioned that network intensive applications are emerging, offering services by interconnecting users to remote IT resources which are distributed across the network (HDTV, multiplayer online gaming ...). Consequently, there is a high need for end-to-end service provisioning algorithms that efficiently provide the required computing and associated network resources in an on-demand fashion.

- **Resiliency** - This is the ability of a system to recover from infrastructure or system faults such as cable cuts or power outages. The business continuity and service availability is seen as the number one obstacle for cloud computing by the authors of [14]. Consequently, a grid/cloud infrastructure should employ specialized hardware and software techniques (protection/recovery) to recover from both network and IT resource failures.

- **Energy considerations of the complete infrastructure** - In order to reduce the carbon footprint and the associated energy budget, service providers are looking for ways to reduce the energy consumption of their infrastructure. Lowering the carbon footprint is possible in two ways: (i) using renewable energy sources such as solar and wind energy and (ii) handling the requests in a way the energy consumption of the complete infrastructure is minimal. This latter option is enabled by devising an energy-aware network and IT service provisioning approach, in which the energy optimization objective is sought during the dynamic allocation of resources.

1.3 Organization of this work

The main objective of this work is to address the grid/cloud issues described in Section 1.2. Optical grids and clouds are described in Chapter 2. Chapter 3 provides a description of a simulation environment which has been developed to evaluate the proposed solutions addressing the energy and scalability issues described above

In Chapter 4 we address the issue of resiliency against single link network failures and show how the anycast routing principle, which is typical of grids/clouds, can be exploited in providing efficient shared path protection. Indeed, as users generally allow the grid/cloud system to decide upon the location where their service is executed, we allow relocation to alternate backup server sites in case of failures. We investigate two different integer linear program (ILP) models for the anycast routing problem, deciding on the primary and backup server locations as well as on the lightpaths towards them. The second model is a large scale optimization

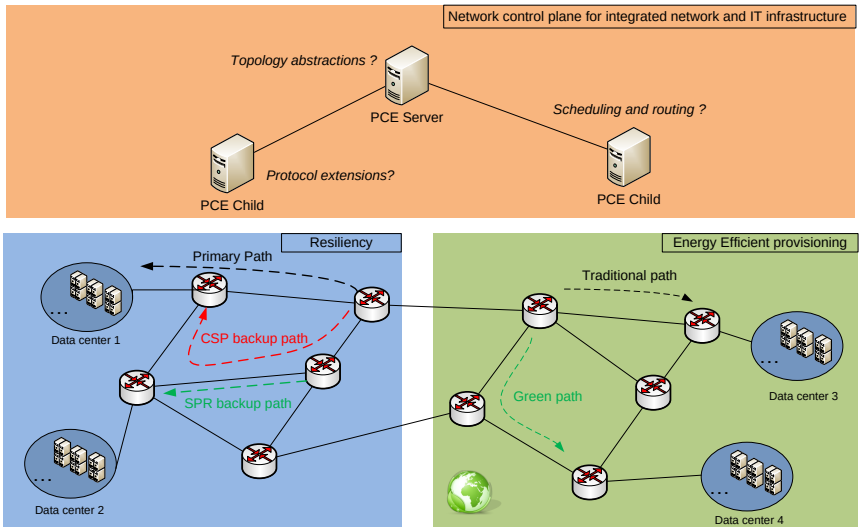


Figure 1.2: This figure shows the contributions made throughout this thesis. The upper part indicates the need for a management mechanism, controlling both network and IT resources. The bottom left part demonstrates that relocation requires fewer network resources to ensure service continuity. The bottom right part reflects the routing and scheduling strategy to minimize the infrastructure's energy consumption.

model which can be efficiently solved using the column generation technique. We also design two new heuristics: the first, although providing nearly optimal solutions, lacks scalability, while the second heuristic is highly scalable, at the expense of a reduced accuracy.

We address the issue of energy consumption in an integrated IT and network infrastructure in Chapter 5. We propose a routing and scheduling algorithm for a grid/cloud architecture, which targets minimal total energy consumption by enabling switching off unused network and/or IT resources, exploiting the cloud-specific anycast principle. A detailed energy model for the entire cloud infrastructure comprising a wide area optical network and IT resources is provided. This model is used to make a single-step decision on which IT end points to use for a given request, including the routing of the network connection towards these end points. Our simulations quantitatively assess the energy-efficient algorithm's potential energy savings, but also its influence on traditional quality of service parameters such as service blocking. Furthermore, we compare the one-step scheduling with traditional scheduling and routing schemes, which calculate the resource provisioning in a two-step approach (selecting first the destination IT end point, and subsequently unicast routing towards it).

Chapter 6 proposes an enhanced network control plane, the NCP+, which is

based on a GMPLS control plane with a Hierarchical Path Computation Element (PCE) architecture to jointly make anycast network routing and IT server provisioning decisions. We discuss (i) the architecture of the NCP+, (ii) two IT-aware aggregation mechanisms to be used in the hierarchical PCE approach and (iii) routing and scheduling algorithms for those aggregation mechanisms.

Finally, Chapter 7 summarizes the contributions, concluding the work while also suggesting future work to optimize optical grids/clouds even further.

1.4 Publications

If authors regard it as essential to indicate that two or more co-authors are equal in status, they may be identified by an asterisk symbol with the caption "These authors contributed equally to this work" immediately under the address list.

1.4.1 Publications in international journals (listed in the Science Citation Index[1])

1. De Leenheer, M.; Develder, C.; **Buysse, J.**; Dhoedt, B. and Demeester, P., *Performance analysis and dimensioning of multi-granular optical networks*, Opt. Switch. and Netw.,Vol.6(2),pp. 88-98, 2009

2. **Buysse, J.**; De Leenheer, M.; Dhoedt, B. and Develder, C., *Providing resiliency for optical grids by exploiting relocation: A dimensioning study based on ILP*, Comput. Commun.,Vol. 34(12), pp. 1389-1398, 2011

3. Shaikh, A.*; **Buysse, J.***; Jaumard, B. and Develder, C., *Anycast routing for survivable optical grids: scalable solution methods and the impact of relocation*, IEEE/OSA J. Opt. Commun. Netw., Vol. 3(9), pp. 767-779, 2011 (* These authors contributed equally to this work)

4. Belter, B.; Ignacio Aznar, J.; Rodriguez Martinez, J.; Contreras, L. M.; Antoniak-Lewandowska, M.; Buffa, G.; **Buysse, J.**; Demchenko, Y.; Donadio, P.; Drzewiecki, L.; Escalona, E.; Garcia Espin, J. A.; Gheorghiu, S.; Ghijsen, M.; Gutkowski, J.; Landi, G.; Parniewicz, D.; Ferrer Riera, J.; Robinson, P. and Soudan, S., *The GEYSERS optical test-bed: a platform for the integration, validation and demonstration of cloud-based infrastructure services*, submitted to Comp. Netw., 2012

[1]The publications listed are recognized as 'A1 publications', according to the following definition used by Ghent University: A1 publications are articles listed in the Science Citation Index, the Social Science Citation Index or the Arts and Humanities Citation Index of the ISI Web of Science, restricted to contributions listed as article, review, letter, note or proceedings paper.

5. Develder, C.; **Buysse, J.**; Dhoedt, B. and Jaumard, B. *Joint dimensioning of server and network infrastructure for resilient optical grids/clouds*, submitted to IEEE/ACM Trans. on Netw.

6. **Buysse, J.**; Georgakilas, K.; Tzanakaki, A.; De Leenheer, M.; Dhoedt, Bart and Develder, C.; *Energy-efficient resource provisioning algorithms for optical clouds*, IEEE/OSA J. Opt. Commun. Netw., Vol. 5(3), pp. 226-239, 2013

7. **Buysse, J.**; De Leenheer, M.; Develder, C.; Miguel Contreras, L. and Landi, G.; *NCP+: An integrated network and IT control plane for cloud computing*, submitted to J. Opt. Switch. and Netw.

1.4.2 Publications in book chapters

1. Vicat-Blanc, P.; Figuerola, S.; Chen, X.; Landi, G.; Escalona, E.; Develder, C.; Tzanakaki, A.; Demchenko, Y.; Garca-Espn, J. A.; Ferrer, J.; Lpez, E.; Soudan, S.; **Buysse, J.**; Jukan, A.; Ciulli, N.; Brogle, M.; van Laarhoven, L.; Belter, B.; Anhalt, F.; Nejabati, R.; Simeonidou, D.; Ngo, C.; de Laat, C.; Biancani, M.; Roth, M.; Donadio, P.; Jimnez, J.; Antoniak-Lewandowska, M. and Gumaste, A., *Bringing optical networks to the cloud: an architecture for a sustainable future Internet*, in The Future Internet, Springer, Vol. 6656/2011,pp. 307-320, 2011

1.4.3 Publications in international conferences (listed in the Science Citation Index[2])

1. De Leenheer, M.; Develder, C.; **Buysse, J.**; Dhoedt, B. and Demeester, P., *Dimensioning of combined OBS/OCS networks*, Proc. 5th Int. Conf. on Broadband Commun., Netw. and Systems (Broadnets), London, UK, 8-11 Sep. 2008

2. De Leenheer, M.; Develder, C.; Vermeir, J.; **Buysse, J.**; De Turck, F.; Dhoedt, B. and Demeester, P., *Performance analysis of a hybrid optical switch*, Proc. Int. Conf. on Opt. Netw. Design and Modelling (ONDM), Vilanova i la Geltru, Spain, 12-14 Mar. 2008

3. **Buysse, J.**; De Leenheer, M.; Develder, C.; Dhoedt, B. and Demeester, P., *Cost-effective burst-over-circuit-switching in a hybrid optical network*, Proc. 5th Int. Conf. Netw. and Services (ICNS), Valencia, Spain, 20-25 Apr. 2009

[2]The publications listed are recognized as 'P1 publications', according to the following definition used by Ghent University: P1 publications are proceedings listed in the Conference Proceedings Citation Index - Science or Conference Proceedings Citation Index - Social Science and Humanities of the ISI Web of Science, restricted to contributions listed as article, review, letter, note or proceedings paper, except for publications that are classified as A1.

4. **Buysse, J.**; De Leenheer, M.; Dhoedt, B. and Develder, C.; *Exploiting relocation to reduce network dimensions of resilient optical Grids*, Proc. 7th Int. Workshop Design of Reliable Commun. Netw. (DRCN), Washington D.C., USA, 25-28 Oct. 2009

5. **Buysse, J.**; Develder, C.; Leenheer, M. D. and Dhoedt, B., *Dimensioning resilient optical Grids*, Proc. 5th Reliability Issues in Next Gen. Opt. Netw. Workshop (RONEXT), co-located with 11th Int. Conf. on Transp. Opt. Netw. (ICTON), Island of Sao Miguel, Portugal, 28 Jun.-2 Jul. 2009

6. De Leenheer, M.; **Buysse, J.**; Develder, C.; Dhoedt, B. and Demeester, P., *Deflection routing in anycast-based OBS grids*, Proc. Int. Workshop on Opt. Burst/Packet Switching (WOBS), in conjunction with Broadnets, Madrid, Spain, 14-17 Sep. 2009

7. **Buysse, J.**; De Leenheer, M.; Dhoedt, B. and Develder, C., *On the impact of relocation on network dimensions in resilient optical Grids*, Proc. 14th Int. Conf. on Opt. Netw. Design and Modelling (ONDM), Kyoto, Japan, 31 Jan.-2 Feb. 2010 (*Travel Grant Award*)

8. Jaumard, B.; **Buysse, J.**; Shaikh, A.; De Leenheer, M. and Develder, C., *Column generation for dimensioning resilient optical grid networks with relocation*, Proc. IEEE Global Telecommun. Conf. (GLOBECOM), Miami, USA, 6-10 Dec. 2010

9. **Buysse, J.**; Cavdar, C.; De Leenheer, M.; Dhoedt, B. and Develder, C., *Improving energy efficiency in optical cloud networks by exploiting anycast routing*, Proc. Asia Commun. and Photonics Conf. (ACP), Shangai, China, 13-16 Nov. 2011

10. **Buysse, J.**; Georgakilas, K.; Tzanakaki, A.; De Leenheer, M.; Dhoedt, B.; Demeester, P. and Develder, C., *Calculating the minimum bounds of energy consumption for cloud networks*, Proc. IEEE Int. Conf. Comp. Commun. and Netw. (ICCCN), Maui, Hawaii, USA, 31 Jul.-4 Aug. 2011

11. De Leenheer, M.*; **Buysse, J.***; Mets, K.; Dhoedt, B. and Develder, C., *Design and implementation of a simulation environment for network virtualization*, Proc. 16th IEEE Int. Workshop Comp. Aided Modeling, Analysis and Design of Commun. Links and Netw. (CAMAD), Kyoto, Japan, 10-11 Jun. 2011 (* These authors contributed equally to this work)

12. Develder, C.; **Buysse, J.**; Shaikh, A.; Jaumard, B.; De Leenheer, M. and Dhoedt, B., *Survivable optical grid dimensioning: anycast routing with server and network failure protection*, Proc. IEEE Int. Conf. Commun. (ICC), Kyoto, Japan, 5-9 Jun. 2011

13. Escalona, E.; Peng, S.; Nejabati, R.; Simeonidou, D.; Garca-Espn, J. A.; Ferrer, J.; Figuerola, S.; Landi, G.; Ciulli, N.; Jimenez, J.; Belter, B.; Demchenko, Y.; de Laat, C.; Chen, X.; Yukan, A.; Soudan, S.; Vicat-Blanc, P.; **Buysse, J.**; De Leenheer, M.; Develder, C.; Tzanakaki, A.; Robinson, P.; Brogle, M. and Bohnert, T. M., *GEYSERS: A novel architecture for virtualization and co-provisioning of dynamic optical networks and IT services*, Proc. Future Netw. Mobile Summit, Lisbon Portugal, 3-5 Jul. 2011

14. Tzanakaki, A.; Anastasopoulos, M.; Georgakilas, K.; **Buysse, J.**; De Leenheer, M.; Develder, C.; Peng, S.; Nejabati, R.; Escalona, E.; Simeonidou, D.; Ciulli, N.; Landi, G.; Brogle, M.; Manfredi, A.; Lopez, E.; Ferrer Riera, J.; Garcia-Espin, J. A.; Figuerola, S.; Donadio, P.; Parladori, G. and Jimenez, J., *Energy efficiency considerations in integrated IT and optical network resilient infrastructures (Invited)*, Proc. 13th Int. Conf. Transparent Opt. Netw. (ICTON), Stockholm, Sweden, 26-30 Jun. 2011

15. Tzanakaki, A. ; Anastasopoulos, M.; Georgakilas, K.; **Buysse, J.**; De Leenheer, M.; Develder, C.; Peng, S.; Nejabati, R.; Escalona, E.; Simeonidou, D.; Ciulli, N.; Landi, G.; Brogle, M.; Manfredi, A.; Lopez, E.; Ferrer Riera, J.; Garca-Espn, J.; Donaldio, P.; Parladori, G. and Jimenez, J., *Energy efficiency in integrated IT and Optical Network infrastructures: The GEYSERS approach*, Proc. IEEE Infocom Workshop on Green Commun. Netw., Shanghai, China, 10-15 Apr. 2011,

16. De Leenheer, M.; **Buysse, J.**; Develder, C. and Mukherjee, B., *Isolation and resource efficiency of virtual optical networks*, Proc. Int. Conf. Comp., Netw. and Commun. (ICNC), Maui, Hawaii, USA, 30 Jan. - 2 Feb. 2012

17. Ferrer Riera, J.; Garcia-Espin; Figuerola, S.; **Buysse, J.**; De Leenheer, M.; Develder, C. and Peng, S., *Converged IT and optical network virtualisation: the Last (clean) step towards the future internet infrastructure management*, Proc. Eur. Conf. on Netw. and Opt. Commun., in conjunction with 4th Conf. on Opt. Cabling and Infr. (NOC/OC&I), Vilanova i la Geltru, Spain, 20-22 Jun. 2012

18. Landi, G.; Ciulli, N.; **Buysse, J.**; Georgakilas, K.; Anastasopoulos, M.; Tzanakaki, A.; Develder, C.; Escalona, E.; Parniewicz, D. and Binczewski, Arthur Stroiski, M., *A network control plane architecture for on-demand co-provisioning of optical network and IT services*, Future Network & Mobile Summit, Berlin, Germany, 4-6 Jul. 2012

1.4.4 Publications in other international conferences

1. De Leenheer, M.; Develder, C.; **Buysse, J.**; Dhoedt, B. and Demeester, P., *Dimensioning of combined OBS/OCS networks*, Proc. Int. Workshop on Optical Burst/Packet Switch. (WOBS), London, UK, 8 Sep. 2008

2. De Leenheer, M.; **Buysse, J.**; Develder, C.; Dhoedt, B. and Demeester, P., *Design of multi-granular optical networks*, Proc. Eur. Conf. on Netw. and Opt. Commun., in conjunction with 4th Conf. on Opt. Cabling and Infr. (NOC/OC&I),Valladolid, Spain, 10-12 Jun. 2009,

3. **Buysse, J.**; Develder, C.; De Leenheer, M. and Dhoedt, B., *ILP and scalable heuristics for dimensioning resilient optical grids*, Proc. 14th INFORMS Telecommun. Conf., Montreal, Canada, 5-9 May 2010

4. A. Tzanakaki, M. Anastasopoulos, K. Georgakilas, **J. Buysse**, M. De Leenheer, C. Develder, S. Peng, R. Nejabati, E. Escalona, D. Simeonidou, N. Ciulli, G. Landi, M. Brogle, A. Manfredi, E. Lopez, J. F. Riera, J. A. Garcia-Espin, S. Figuerola, P. Donadio , G. Parladori, J. Jimenez, A. T. De Duenyas, P. Vicat-Blanc, J. van der Ham, C. de Laat, M. Ghijsen, B. Belter, A. Binczewski, M. Antoniak-Lewandowska, *Power considerations for ICT sustainability: the GEYSERS approach*, Proc. 4th Future Internet Cluster Topic Workshops on ICT and Sustainability, Budapest, Hungary, 16 May 2011

5. Aznar, J. I.; Rodriguez, J.; **Buysse, J.**; Peng, S.; Anhalt, F. and Garca-Espn, J. A., On the viability of a CSO Architecture for on-demand virtualized Cloud Provisioning and Planning, IEEE Int. Conf. on Cloud Networking, Paris, France, 28-30 Nov. 2012

6. Develder, C.; **Buysse, J.**; De Leenheer, M.; Jaumard, B. and Dhoedt, B., *Resilient network dimensioning for optical grid/clouds using relocation (Invited Paper)*, Proc. Workshop on New Trends in Opt. Netw. Survivability, at IEEE Int. Conf. on Commun. (ICC), Ottawa, Ontario, Canada, 11 Jun. 2012

7. Garcia-Espn, J. A.; Ferrer Riera, J.; Figuerola, S.; Ghijsen, M.; Demchemko, Y.; **Buysse, J.**; De Leenheer, M.; Develder, C.; Anhalt, F. and Soudan, S., *Logical infrastructure composition layer, the GEYSERS holistic approach for infrastructure virtualisation*, Proc. TERENA Netw. Conf. (TNC),Reykjavk, Iceland, 21-24 May 2012

1.4.5 Publications in national conferences

1. **Buysse, J.**; *Anycast routing: design using column generation*, 11th UGent-FirW PhD symposium, Ghent, Belgium, 2010

References

[1] G. Coulouris, J. Dollimore, and T. Kindberg. *Distributed Systems Concepts and Design*. Addison Wesley, 2005.

[2] *Search for extraterrestrial intelligence*. Available from: http://setiathome.berkeley.edu/.

[3] *HPC UGent*. Available from: http://www.ugent.be/hpc/en.

[4] C. Kesselman and I. Foster. *The grid: blueprint for a new computing infrastructure*. Morgan Kaufmann Publishers, Nov. 1998.

[5] R. Pordes, D. Petravick, B. Kramer, D. Olson, M. Livny, A. Roy, P. Avery, K. Blackburn, T. Wenaus, F. Wrthwein, I. Foster, R. Gardner, M. Wilde, A. Blatecky, J. McGee, and R. Quick. *The open science grid*. Journal of Physics: Conf. Series, 1(1):12–57, 2007.

[6] D. Reed. *Grids, the TeraGrid and beyond*. IEEE Computer, 78(1):12–57, Jan. 2003.

[7] O. Appleton, B. Jones, D. Kranzlmüller, and E. Laure. *The EGEE-II project: evolution towards a permanent european grid initiative*. In Advances in Parallel Comp. : High Perf. Comp. and Grids in Action, volume 16, pages 424–435. KTH, Centre for High Performance Computing, 2008.

[8] N. Heath. *Cloud computing to save tech budgets*, Jan. 2011. Available from: http://www.techrepublic.com/blog/cio-insights/cloud-computing-to-save-tech-budgets/39746847.

[9] R. Mullins. *Microsoft study shows 80% savings by using the cloud*, Oct. 2011. Available from: http://www.networkworld.com/community/blog/microsoft-study-show-80-savings-using-cloud.

[10] *Sales Cloud*. Available from: http://www.salesforce.com/eu/?ir=1.

[11] *Amazon Elastic Compute Cloud*. Available from: http://aws.amazon.com/ec2/.

[12] *Amazon Web Services*. Available from: http://aws.amazon.com/.

[13] R. Rotem-Gal-Oz. *Fallacies of distributed computing explained*, May 2006. Available from: http://www.rgoarchitects.com/Files/fallacies.pdf.

[14] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia. *A view of cloud computing*. ACM Commun., 53(4):50–58, Apr. 2010.

[15] P. Hyek, Y. Munakata, R. Norris, J. Tsang, C. Flynn, K. Price, and J. Steger. *Cloud computing issues and impacts*. Global Technology Industry Discussion Series, 1(1):1–56, Apr. 2011.

[16] L. Garber. *Converged infrastructure: Addressing the efficiency challenge*. IEEE Computer, 45(8):17–20, Aug. 2012.

2

Introduction to optical clouds and grids

"[...] Comes from the early days of the Internet where we drew the network as a cloud ... We did not care where the messages went, the cloud hid it from us"

–Kevin Marks, Google

2.1 The need for grid and cloud computing

Imagine a computer scientist who has built a simulator application and wants to evaluate an algorithm he had just created. Assume that in general, a simulation takes about one hour and that he needs 11 data points for his study. In order to be statistically significant, suppose each data point needs to be replicated 20 times (to get substantial confidence intervals as randomness is involved in the simulation). If he performed each simulation on his own desktop, this would require $11 * 20 = 220$ hours of computing (and waiting) time. However, if he transformed each simulation into a job which he offloads to a system which performs these 220 simulations in parallel, he would only require 1 hour. This is one example application of grid/cloud computing, which we have used many times throughout this work (together with a lot of reruns due to unforeseen programming bugs).

In general, there are three main categories in which applications, making use of a form of distributed computing, can be divided.

1. Scientific applications (or e-Science)
2. Consumer applications

Table 2.1: Examples of e-Science projects and initiatives.

Name	Description
AstroGrid	AstroGrid is a software suite which enables astronomers to access a Virtual Observatory (VO): the astronomers have access to all sorts of astronomical data on which scientific analyses can be performed [1].
Climate-G	This project has built a distributed environment, using grid and P2P technologies, for scientists to carry out geographical and cross-institutional discovery, access, visualization and sharing of climate data. [2]
Flow Grid	This is an example of a project which aims to enable computational fluid dynamics (CFD) (problems that involve fluid flows) simulations to be set-up, executed and monitored on geographically and organizationally dispersed computing resources. [3]
LHC Computing Grid	This is an international initiative which has built a grid computing infrastructure to store, distribute and analyze the ca. 25 petabytes of data annually generated by the Large Hadron Collider (LHC) at CERN on the Franco-Swiss border [4]. It uses (and is part of) the infrastructure provided by the Enabling Grids for E-sciencE project [5].
BioinfoGRID	This project [6] evaluates applications in the fields of genomics, proteomics, transcriptomics and drug discovery, using the infrastructure provided by the EGEE project.

3. Business applications

In what follows we will discuss these categories and provide some real world examples.

Scientific applications: Science applications (particularly e-Science) reflect computationally intensive workloads which generally produce a large amount of data and require a lot of processing time. These applications are organized as a set of interdependant tasks (dependending on each other) and can run in parrallel in a distributed environment. Moreover, these tasks sometimes require communication to obtain intermediate results. Areas of e-Science which benefit from distributed computing include astrophysics, weather forecast, computational fluid dynamics, high-energy physics, computational biology, etc. Some example projects are described in Table 2.1.

Consumer applications. The emergence of the World Wide Web (WWW) has provided the ordinary computer user access to different kinds of distributed com-

puting systems. This in turn has led to a shift in use of the Internet: the paradigm of simple web browsing has changed into a more attractive and interactive use of the web. Consider applications such as Youtube, Flicker, and Facebook, which are very demanding in terms of data storage. Youtube for example, states that every minute about 60 hours of video is being uploaded [1]. If we assume a reference video of 10 min with a size of 100MB, this means we have 35 GB of uploaded data per minute, 49 TB of uploaded data per day and 18PB of uploaded data per year which needs to be stored.

Not only web applications have opened the gate for the consumer, also multiplayer gaming (e.g., World of Warcraft [7]) and to a lesser extent, augmented reality, have come into play. These applications have more stringent requirements on the (immediate) processing and delivery of the data as the game/application should respond in sync with the player's controls.

Business applications. Distributed computing has made its entrance in the business community for some time now. Exemplary are data mining companies which apply several algorithms to find connections and structures in data sets to create commercial profiles. Not only private companies benefit from distributed computing, the Flemish public broadcasting (VRT) has investigated the use of a grid environment in media production. An example is the grid based transcoding of videos into a lower rate format, so they can be offered via their web site [8].

Cloud computing has also been launched in the business community. Instead of spending a large capital cost for installing a private IT infrastructure, companies outsource this to specialized firms. This has several advantages:

- Capex reduction.
- Opex reduction.
- Seamingly infinite access to computational/storage resources.
- Automated backup and recovery.
- IT infrastructure scales on demand.
- Easy access to data, as you can access it anywhere (e.g., web access).

Cloud computing brings a lot of benefits on the table, but cannot overlook possible hidden hazards. As the company is surrendering its valuable information to the cloud provider, it is essential that it is trustworthy and that security is in place to protect the data from malicious attacks. Moreover, as no system is foolproof, situations will inevitably occur where either network connectivity or computer problems arise. Consequently, the cloud provider needs to assure the

[1] http://www.youtube.com/t/press_statistics

client that its offered service is resilient and is able to recover from failures in a fast time frame.

In conclusion, we can state that a large set of applications can benefit from the advantages of grid and cloud computing infrastructures. Based on the examples provided above, we see that these architectures have several requirements:

- On demand instantiation of (geographically spread) resources.
- Fast response time and low latency (for interactive and real-time services).
- Ability to support a large amount of network and computational resources.
- Adaptation to the particular resource needs over time for an application (elasticity).
- Reliable services with fast restoration in case of failures.

2.2 Architecture and models

In this section we will formally discuss the definitions for grid and cloud computing.

2.2.1 Grid Computing

In 1998 Carl Kesselman and Ian Foster, the founders of the grid concept, defined the grid as follows in their book "The Grid: Blueprint for a New Computing Infrastructure" [9]:

> "A computational grid is a hardware and software infrastructure that provides dependable, consistent, pervasive and inexpensive access to high-end computational capabilities."

This very broad definition was later refined in [10] to the following :

> "Grid computing is concerned with coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations (VOs)."

From the definitions above, we can derive the following check list (a superlist of the one described in [11]) of properties a system must satisfy before it can be called a grid infrastructure.

☑ **Resources.** A grid consists of several heterogeneous resources, which can be divided into three classes: (i) computational resources (e.g., PCs, clusters) which are characterized by processing speed and memory, (ii) storage

Figure 2.1: A grid infrastructure consists of heterogeneous, geographically spread resources which are not subject to centralized control.

resources offering storage capabilities (e.g., databases) or information generation equipment (e.g., sensor equipment) and (iii) networking equipment such as nodes and links.

☑ **Geographically dispersed.** Given the heterogeneity of the resources, it is no surprise that these resources are distributed and within different control domains (companies, universities, etc.).

☑ **No centralized control.** The resources are not subject to centralized control, as we have previously indicated that they reside within different control domains. This is a fundamental difference with cluster computing, where the resources are always in the same control and administrative domain.

☑ **Using standards, general protocols and interfaces.** A grid is built from multi-purpose, standard, open protocols and interfaces to address different functionalities such as authentication, resource access, etc. Consequently, resource-sharing arrangements can be set up dynamically with any party, as opposed to an application-specific system.

☑ **We achieve a higher performance by combining the consituents into one**

architecture. As sharing of resources is possible between the different constituents, the system should be able to meet specific user demands which were impossible to address without access to the resources in the other control domains.

As a last note, we want to indicate that there are two main classes of grid architectures, depending on the sort of applications they are supporting.

1. Computational grid. Such a system offers a huge number of fast computational resources, to tackle complex problems. An example is BioinfoGRID [6].

2. Data grid. Their main focus lies in the distributed data management, analyzing and storing different data sets from different providers (e.g., AstroGrid) [1].

2.2.2 Cloud Computing

Many definitions of cloud computing have been proposed [12–16] which together with a variety of supporting technologies have led to a lot of confusion. The memorable quote on the term cloud computing of Lary Ellison, CEO of Oracle corporation, illustrates this well:

> "The computer industry is the only industry that is more fashion-driven than women's fashion. [...] But I don't understand what we would do differently in the light of cloud."

In order to avoid this kind of confusion, we provide a check list (the same way we formalized the grid definition) of properties a computing system must possess, before it can be denoted as a cloud infrastructure.

☑ **Resources.** Analogous to grid computing, cloud computing offers resources to its users. This is performed automatically, without requiring human interaction with each service provider. In contrast to grid resources, clouds cover a wider scope: applications, platforms and the infrastructure itself can be provided as a service, leading to the Anything-as-a-service (XaaS) paradigm (where X is Software, Platform or Infrastructure - see Section 2.2.2.1).

☑ **Location independent.** There is a sense of location independance, as the user has no knowledge over the exact location of the provided resources but may be able to specify a location at a higher level of abstraction (e.g., country, state, or datacenter).

- ☑ **Broad network access.** Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, tablets, laptops, and workstations).
- ☑ **User-friendliness.** As cloud infrastructures are targeted to a wider audience (i.e., business and consumers as opposed to just specialized scientists), the deployment, configuration and access to the offered resources should be easy and straightforward. In most cases, this is done by offering a website where all actions can be performed.
- ☑ **Virtualization** is the concept where virtual (rather than actual) versions of something are created such as an operating system (OS), storage devices or networking equipment. While a physical resource (such as a server or hard disk) is clearly an actual device, both *logically* (from the user's point of view) and *physically* (from a hardware perspective), a virtual machine is *logically* a complete machine, but *physically* merely is a set of files and running programs on an actual physical machine. Consquently, a physical machine can support multiple concurrent virtual machines which allows a cloud provider to use statistical multiplexing[2] to overcome overprovisioning to cope with peak loads. This virtualization also allows migration (or relocation) of applications to other servers to boost performance or recover from failures.
- ☑ **Scalability** is one of the main drivers for the success of cloud computing. The property of a cloud to upscale/downscale IT requirements with the IT demand, is very attractive for both small and large companies. Hence, consumers are typically billed on a pay-as-you-use basis as opposed to making large capital investments in the necessary hardware upfront.
- ☑ **Use of Service-Level-Agreements (SLAs),** are negotiated agreements between the provider and the users, detailing the agreed understanding about responsibilities, guarantees, and warranties of the services offered by the provider.

Setting out from this checklist, we can define cloud computing as follows:

> Cloud computing is an architecture which provides easy access over a network to a large pool of (virtualized) resources (hardware or software), which can be easily (re)configured to cope with the offered load on a pay-as-you-use basis. The properties of the offered resources are guaranteed by means of SLAs.

[2] sharing of a medium with multiple users

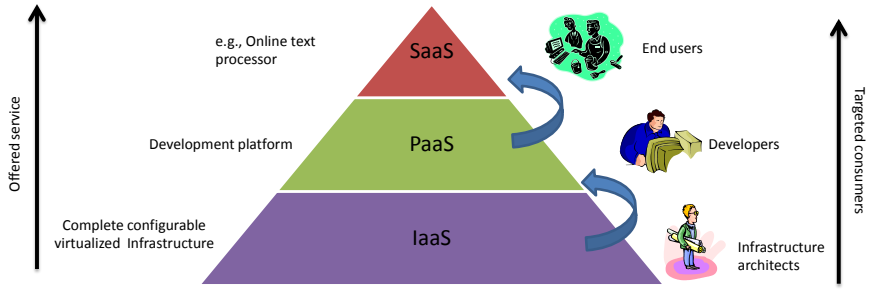


Figure 2.2: Cloud computing provides easy access to a large pool of resources (hardware or software), which can be easily (re)configured to cope with the offered load on a pay-as-you-use basis. The properties of the offered resources are guaranteed by means of SLAs.

2.2.2.1 Different forms of cloud computing

Depending on the offered service of the cloud provider, there are three cloud scenarios, which are generally referred to as Anything-as-a-Service (XaaS - see Fig. 2.2). In the subsequent sections, we will discuss them bottom-up.

Infrastructure-as-a-Service (IaaS). The cloud provider owns a large pool of resources (computing, storage and network) which are virtualized and offered to the users as an ad-hoc infrastructure. The virtualization of the resources allows the provider to aggregate or decompose resources and dynamically resize the offered infrastructure. An example of a company providing IaaS is Amazon EC2 [17]. IaaS is also exemplified by the GEYSERS project, as presented in Chapter 3, Section 3.3.

Platform-as-a-Service (PaaS). By offering an extra abstraction level on top of IaaS, the cloud provider could opt to provide a software platform on top of the virtualized infrastructure. Consumers are provided with a development platform on which they can generate their own applications, while the scaling of the infrastructure under influence of the application is performed transparently. Examples are Amazon Web Service [18] , Google Apps Engine [19] and Windows Azure [20].

Software-as-a-Service (SaaS). Finally, software as a resource can be found as the highest layer in cloud computing. Here, applications are provided as services and run remotely (in the cloud) as opposed to running them locally. Typical applications are office programs such as word and data processors. Example SaaS providers are SalesForce [21] and Micrsoft Online Services [22].

Grids and clouds aspire the same ambition: reducing computing costs and increasing flexibility and scalability by using third-party hardware and software. It

is clear that from a conceptual point of view, clouds and grids share a lot of the same features, and that the cloud infrastructure can be seen as the next step of grid evolution, targeting the business audience (as opposed to the science community). With the introduction of web access to grids (see [23] for instance), it is only a matter of time for the concepts and architectures for cloud and grids to converge. For a more detailed and general comparison between grid and cloud architectures we refer to [24, 25].

2.3 Underlying transport architectures

From the example applications given in Section 2.1 and the definitions given for grid and cloud computing, it is clear that there is a need for a network connecting all the data centers which are housing the computational resources. Requirements necessary for a fluent and efficient operation of grid and clouds include low latency (as services must be provided on demand) and high bandwidth (a huge amount of data needs to be dealt with) connections. This is exactly what photonic networking offers :

- **Capacity.** Fiber optics offer high bandwidth connections (experimental speed up to 69.1 Tb/s [26]), much higher than coaxial cabling (around 125 Mbits/s).
- **Dependable.**
  - Low bit error rates (typically around 10^{-12}).
  - Less signal degradation than in copper wire (fiber loses only 3% while copper cable 94% of its original signal strength over 100 meters).
  - Light signals in one fiber do not interfere with those from other fibers.
  - Imperviousness to electrical noise, as it does not use an electrical connection.
- **Secure.** While tapping optical fibers is possible, it is difficult and results in additional loss which is easily detectable.
- **Cost.** Fibers are less expensive than its equivalent lengths of copper wire:
  - Increased capacity of optical fiber means that producing fibers cost substantially less than producing copper wires.
  - As fiber is smaller and lighter than copper cabling which means more fiber can fit a wiring duct than its copper counterpart.

  Moreover lower-power transmitters can be used instead of the high-voltage electrical transmitters needed for copper wires.

When the core network of a grid/cloud infrastructure is an optical network, we use the term *optical grid/cloud computing*.

2.3.1 Optical transmission

An optical network consists of links and nodes. The basic functionality of a network is to establish connections between nodes. The route followed by the connection is denoted as the light path between the nodes, which consists of the reserved wavelengths from the links along that path. The end points of such a path are called the terminal nodes, which send and receive information. The function of the intermediate nodes, is to direct incoming traffic to the correct outgoing link. In photonic networks, the nodes are called optical cross-connects (OXC). The optical fiber (made of silica) is the transmission medium for optical networks. These fibers serve as a light pipe, where light can pass from one end of the fiber to the other end.

There are two ways to support multiple data streams on a fiber: (i) Time Division Multiplexing (TDM) and (ii) Wavelength Division Multiplexing (WDM) which can be used in a combination by time-division multiplexing fixed slots onto a specific wavelength. In TDM, the wavelength (the light signal) is divided into different time slots which are assigned to each data stream. WDM however, multiplexes a number of optical carrier signals (with a multiplexer, MUX) onto a single fiber by using different wavelengths (i.e., colors) of laser light, as shown in Fig. 2.3. This optical signal is then transmitted over the optical medium and regenerated where necessary (i.e., after a certain fiber span). At the end of the tunnel, the demultiplexer (DEMUX) separates the individual wavelength components. WDM systems are divided into two categories: (i) Coarse WDM (CWDM) which provides up to 8 channels per fiber and (ii) Dense WDM (DWDM) which typically supports 40-80 channels per fiber. CWDM cannot be amplified by an Erbium-Doped Fiber Amplifier (EDFA) which limits is fiber span to 60 km for a 2.5 Gbit/s signal (suitable for e.g., metropolitan areas). DWDM however can be amplified by EDFAs increasing the distance a light signal can travel. Hence, DWDM is the perfect candidate to support the core network for grids and clouds.

As already stated above, optical technology is an ideal candidate for core networks. We note however, that it is also applied in access and local networks. For access networks, Passive Optical Networking (PON) has been adopted. A PON is a point-to-multipoint fiber to the premises network architecture, where splitters and couplers are used for a single fiber to serve multiple users. A PON consists of an optical line terminal (OLT) at the service provider's office and a number of optical network units (ONU) at the user's premises. The downstream data streams are multiplexed on the same fiber using couplers (usually using TDM) by the OLT. Consequently, each ONU receives all data streams and is responsible for demulti-

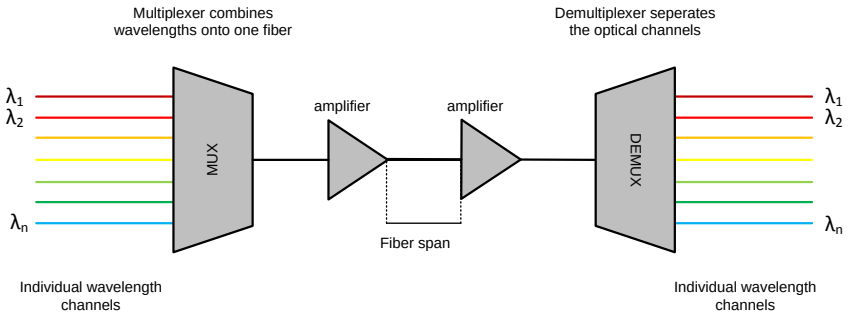


Figure 2.3: Wavelength Division Multiplexing.

plexing its own data stream using a splitter.

The optical LAN is used for instance in very large data centers, which traditionally employed electronic switched networks [27]. However, the methodologies used in the next chapters would remain the same and we note that we do not deal with optical access or LAN architectures.

2.3.2 Introducing the optical cross connect

The second important constituent of an optical network is the optical cross connect (OXC), also refered to as a photonic switching node. Its main functionalities are:

- Extract and insert specific wavelengths from the network. This is performed by an Optical Add-Drop Multiplexer (OADM), which is considered to be a special type of OXC. Moreover, if an OADM is also able to be configured by sofware commands (on top of hardware configuration), we call it a Reconfigurable Optical Add-Drop Multiplexer (ROADM).
- Interconnect incomming fibers to create meshed optical networks.

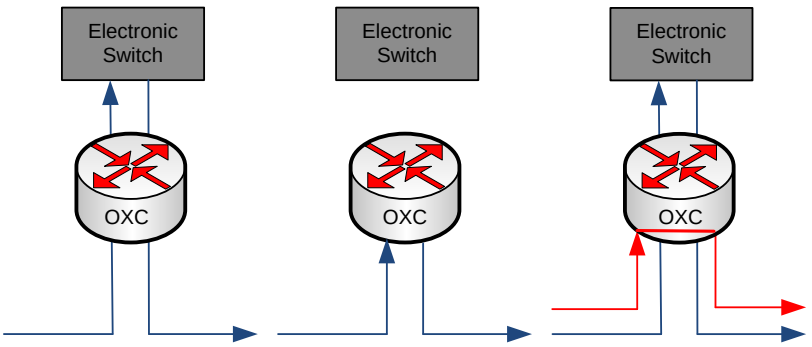
There are three types of switching architectures, as shown in Fig. 2.4

- **Opaque.** This architecture always converts the data stream (after demultiplexing) into the electronic domain. An electronic switching module is responsible for choosing the correct output port. The data is then converted back to the optical domain where it is multiplexed onto the correct outlet optical fiber. The downside of this optical-electronic-optical (OEO) conversion is that (i) it adds an extra delay compared to all-optical switching and (ii) it adds to the associated cost and energy consumption of the switching node. However, the quality of the signal is restored when transformed into the electronic domain, which has the added benefit of recurring signal regeneration. Moreover, the system is able to perform wavelength conversion i.e.,

changing the input wavelength λ_i to another output wavelength $\lambda_j (i \neq j)$. A network where all devices are opaque, is called an opaque optical network.

- **Transparent.** This is the counterpart of the opaque OXC, where the signal stays in the optical domain. The entry wavelength is demultiplexed and switched by an optical switch module (e.g., a wavelength selective switch (WSS)) after which the signals are multiplexed onto the the correct outlet optical fibers. There is no added delay, but quality of the signal is difficult to check. An optical network constituting transparent devices, is denoted as a transparent network.

- **Translucent.** This architecture is the compromise between the opaque and the transparent OXC. There are two switching matrices available, an electronic and a optical one. When a wavelength enters the OXC, it can be either switched by the optical switch (benefiting from the high optical data rate) or by the electronic one (e.g., when conversion or regeneration is necessary). This kind of architecture can leverage the benefits of both architectures. A network which consists of translucent devices is called an translucent optical network.



(a) Opaque. (b) Transparent. (c) Translucent.

Figure 2.4: The different photonic switching architectures. The opaque includes signal regeneration at every intermediate node along a lightpath. The transparent architecture allow signals to bypass extensive electronic signal processing at intermediate nodes. The translucent network allows a signal to remain in the optical domain before its quality degrades, thereby requiring it to be electronically regenerated at an intermediate node.

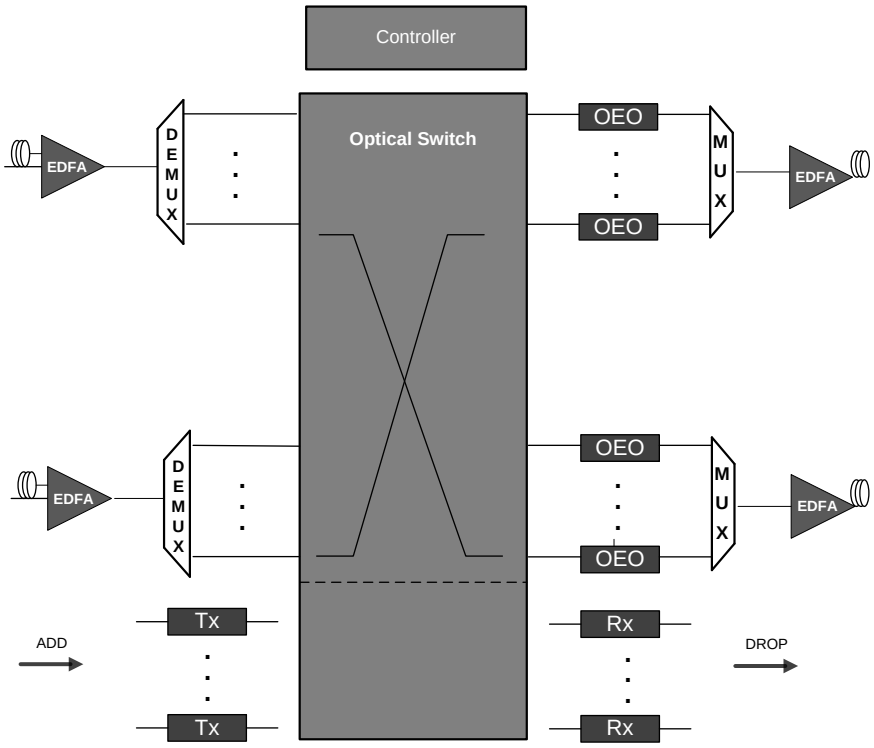


Figure 2.5: Components of a optical cross connect.

2.3.2.1 Optical Components

We mainly used the opaque OXC in our studies which is the architecture mostly deployed by carriers today. An opaque OXC comprises multiple hardware components, which are shown in Fig. 2.5. The first components are the amplifiers at the incoming optical fiber (preamplification). They are used to compensate for the loss (attenuation) in the fiber and increase the intensity of the incoming signal (including noise). Two important amplifier designs are used, namely doped fiber amplifiers (DFA) and Raman amplifiers. After the signal is amplified by the amplifier (an Erbium-DFA or EDFA in Fig. 2.5 which is the most widely used), it enters a demultiplexer which decomposes the entry wavelength spectrum from optical fibers into its constituents (wavelengths). A wavelength then enters the switch matrix, where it is sent to a transponder. The most popular design for a circuit switching matrix is a Wavelength Selective Switch (WSS) based on MEMS (Micro-Electro Mechanical System) technology. The WSS can steer each optical channel present on its input port to one of its output ports according to the wavelength of the channel. The aforementioned transponder is a module which is able

to convert the optical signal to the electronic domain and back. It consists of a receiver and a transmitter component. A transmitter converts an electrical signal into an optical signal while a receiver has the same function but reversed. When the data stream needs to be switched, it is multiplexed after leaving the OEO converter by the multiplexer and amplified by the outgoing EDFA (postamplification). Lastly, we have a controller which is a module responsible for choosing which input channel goes to which output channel.

2.3.3 Optical Switching

Apart from a transmission technology, optical networks adopt a certain switching technology. DWDM networks correspond well to the optical circuit switching (OCS) paradigm, where bandwidth is reserved in advance along a lightpath. The advantage here is that bandwidth is completely reserved and situations will never occur where bandwidth is unavailable. Drawbacks however include the non-negligible setup time for a circuit (which can be overcome by fast circuit switching (FCS) [28]) and the waste of bandwidth if the traffic demand does not match the full capacity of a wavelength. OCS is a mature technology and consequently has been used in several grid and cloud deployments [17, 29, 30].

To overcome the long setup times of optical circuits, optical burst switching (OBS) has been devised as a possible strategy. A user divides its data stream into several chunks which are mapped onto data bursts. Before sending the burst, a header is sent which informs the next hop on the path to reserve bandwidth for the duration of the transmission of the data burst [31]. However, OBS has not been widely adopted yet, mostly due to the challenging hardware requirements involved.

The finest data granularity could be provided by optical packet switching (OPS). The main difference with OBS is that the control information is encapsulated in the packet in OPS, while OBS sends this information before the actual data. However, all optical OPS requires optical memory which has proven to be a big challenge [32]. Optical memory is still a hot research topic with fiber delay lines [33] (based on fiber coils) and optical flip-flop memory [34] to be very promising techniques.

Another promising technique is optical flow switching (OFS) [35]: here users request connectivity for a long period of time ($\geq 100ms$). If a lightpath is granted, the dedicated network resources are relinquished and used for other users. OFS uses an electronic control plane to schedule the connectivity requests. Note that in OFS networks all queuing of data occurs at the end users, thereby obviating the need for buffering in the network core. OFS makes use of statistical multiplexing of large flows by grooming data flows at the edge of the network, to reach a higher utilization as opposed to OCS where each circuit is planned separately.

2.4 Controlling the network

For a photonic network to work, we need a control system that correctly configures the OXC. For optical circuit switched WDM network, GMPLS [36] is the de-facto standard. GMPLS is an extension of MPLS [37], which labels data with information for it to reach its final destination. This information is used in the switching hardware of each router handling the data, which demands less effort from the hardware in comparison with e.g., IP-routing. GMPLS extends this by also supporting TDM, WDM and fiber (port) switching. It is based on a generalized label corresponding to either (i) a single fiber, (ii) a single waveband, (iii) a single wavelength, (iv) or a set of time slots. A path that has been configured by GMPLS is called a Label Switched Path (LSP). GMPLS separates the control plane into three parts: the signaling plane, the routing plane and the link management.

2.4.1 Signaling plane

In order to set up LSPs, a signaling protocol is needed in order to configure the OXCs along the path, distribute the labels and to reserve any of aforementioned resources. The predominant signaling protocol used in GMPLS is the Resource Reservation Protocol with Traffic Engineering extensions (RSVP-TE) [38]. Any actions defined in GMPLS can be performed by this protocol: setup, modify, or remove the LSPs.

2.4.2 Routing plane

To correctly perform resource reservation, allocation, and topology discovery on the available optical link resources, each node needs to maintain a representation of the state of each link in the network. This information includes the number of reserved and available resources or any other valuable information. Open Shortest Path First with Traffic Engineering considerations (OSPF) [39] is typically used to propagate that information.

2.4.3 Link management

GMPLS also uses the Link Management Protocol (LMP) [40] to communicate proper cross-connect information between the network elements. LMP provides control-channel management, link-connectivity verification. Control-channel management establishes and maintains connectivity between adjacent nodes using a keep alive protocol. Link verification evaluates the physical connectivity between nodes, thereby detecting loss of connections and misrouting of cable connections.

2.5 Conclusions

We have started this chapter by identifying the need for distributed computing environments such as grid and cloud computing. We formalized the definitions for the terms "grid" and "cloud" by formulating two checklists of properties, which grids and clouds must satisfy. We concluded that section by indicating the similarities between these architectures, with the statement that grids will probably evolve to some form of cloud computing.

As this thesis deals with optical grids (i.e., the supporting network is a photonic network), we also discussed the optical transmission technology, focussing on circuit switched (D)WDM networks. Lastly we indicated the base functions and associated GMPLS constituents for this kind of photonic network.

References

[1] J. A. Tedds. *Science with the virtual observatory: the Astrogrid VO desktop*. In Proc. Multi wavelength astronomy and the Virtual Observatory, pages 1–8, European Space Astronomy Centre, Spain, Dec. 2008.

[2] G. Aloisio, S. Fiore, S. Denvil, M. Petitdidier, P. Fox, H. Schwichtenberg, H. Blower, and R. Barbera. *The Climate-G testbed: towards a large scale data sharing environment for climate change*. In Proc. EGU General Assembly, pages 1–22, Vienna, Austria, 19 24 Apr. 2009.

[3] A. Andrzejak and J. Wendler. *FlowGrid - flow simulations on-demand using grid computing*. Available from: http://www.unizar.es/flowgrid/download/flowgrid-poster.a4.pdf.

[4] L. Robertson. *From the web to the grid and beyond, computing paradigms driven by high energy physics*, chapter Computing services for LHC: from clusters to grids, pages 69–89. Springer-Verlag, 2012.

[5] F. Gagliardi, B. Jones, F. Grey, M.-E. Begin, and M. Heikkurinen. *Building an infrastructure for scientific grid computing: status and goals of the EGEE project*. Philosophical trans. of the royal society, 363(1833):1729–1742, Aug. 2005.

[6] L. Milanesi. *Guidelines and recommendations for the scientific community based on the experience and the results gained from the BioinfoGRID project*. 2006.

[7] Available from: http://us.battle.net/wow/en/.

[8] *Grid enabled infrastructure for service oriented high definition media applications*. Available from: http://www.iminds.be/en/research/overview-projects/p/detail/geisha.

[9] C. Kesselman and I. Foster. *The grid: blueprint for a new computing infrastructure*. Morgan Kaufmann Publishers, Nov. 1998.

[10] I. Foster, C. Kesselman, and S. Tuecke. *The anatomy of the grid: enabling scalable virtual organizations*. Int. J. High Perform. Comput. Appl., 15(3):200–222, Aug. 2001.

[11] I. Foster. *What is the Grid? - A three point checklist*. GRIDtoday, 1(6):1–3, Jul. 2002.

[12] J. Geelan. *Twenty one experts define cloud computing*. In Cloud Comp. Mag. Cloud-Expo, 2008.

[13] B. De Haaf. *Cloud computing - the jargon is back!* In Cloud Comp. Mag. Cloud-Expo, 2008.

[14] K. A. Delic and M. A. Walker. *Emergence of the academic computing clouds*. Ubiquity, 2008(August):1, Aug. 2008.

[15] R. Buyya. *Market-oriented cloud computing: vision, hype, and reality of delivering computing as the 5th utility*. In Proc. 9th Int. Symp. on Cluster Comp. and the Grid (CCGRID), page 1, Shanghai, China, May 2009.

[16] P. Mell and T. Grance. *The NIST definition of cloud computing*. Technical report, National Institute of Standards and Technology, Information Technology Laboratory, Jul. 2009.

[17] *Amazon Elastic Compute Cloud*. Available from: http://aws.amazon.com/ec2/.

[18] *Amazon Web Services*. Available from: http://aws.amazon.com/.

[19] *Google App Engine*. Available from: https://developers.google.com/appengine/.

[20] *Windows Azure*. Available from: http://www.windowsazure.com/nl-nl/home/features/overview/.

[21] *Sales Cloud*. Available from: http://www.salesforce.com/eu/?ir=1.

[22] *Mircosoft Online Services*. Available from: https://www.aspex.be/nl/ons-aanbod/software-as-a-service/voordelen-van-saas/?gclid=CLqsuuCjgbQCFcbLtAod83QA0g.

[23] N. Wilkins-Diehr, D. Gannon, G. Klimeck, S. Oster, and S. Pamidighantam. *Teragrid science gateways and their impact on science*. Computer, 41(11):32–41, Nov. 2008.

[24] I. Foster, Y. Zhao, I. Raicu, and S. Lu. *Cloud computing and grid computing 360-degree compared*. In Proc. grid Comp. Environ. Workshop, pages 1–10, Nov. 2008.

[25] L. M. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner. *A break in the clouds: towards a cloud definition*. SIGCOMM Comput. Commun. Rev., 39(1):50–55, Dec. 2008.

[26] A. Sano, H. Masuda, T. Kobayashi, M. Fujiwara, K. Horikoshi, E. Yoshida, Y. Miyamoto, M. Matsui, M. Mizoguchi, H. Yamazaki, Y. Sakamaki, and H. Ishii. *69.1-Tb/s (432 x00D7; 171-Gb/s) C- and extended L-band transmission over 240 km Using PDM-16-QAM modulation and digital coherent detection*. In Proc. Opt. Fiber Commun. (OFC), collocated Nat. Fiber Optic Engineers Conf., pages 1 –3, San Diego, CA, USA, March 2010.

[27] N. Farrington, G. Porter, S. Radhakrishnan, H. H. Bazzaz, V. Subramanya, Y. Fainman, G. Papen, and A. Vahdat. *Helios: a hybrid electrical/optical switch architecture for modular data centers*. In Proc. ACM SIGCOMM 2010 conf., SIGCOMM '10, pages 339–350, New York, NY, USA, 2010. ACM.

[28] K.-i. Sato and H. Hasegawa. *Optical networking technologies that will create future bandwidth-abundant networks*. IEEE/OSA Optical Commun. and Netw,, 1(2):81–93, Jul. 2009.

[29] C. T. de Laat and L. Herr. *Ultra high definition media over optical networks (CINEGRID)*. In Proc. Opti. Fiber Commun. Conf., pages 1–20, San Diego, CA, 2226 Mar 2009. Optical Society of America.

[30] T. DeFanti, C. de Laat, J. Mambretti, K. Neggers, and B. St. Arnaud. *TransLight: a global-scale LambdaGrid for e-science*. Commun. ACM, 46(11):34–41, Nov. 2003.

[31] M. De Leenheer, P. Thysebaert, B. Volckaert, F. De Turck, B. Dhoedt, P. Demeester, D. Simeonidou, R. Nejabati, D. Zervas, G. Klonidis, and M. O'Mahony. *A view on enabling consumer oriented grids through optical burst switching*. IEEE Commun. Mag., 44(3):124–131, 2006.

[32] M. O'Mahony, D. Simeonidou, D. Hunter, and A. Tzanakaki. *The application of optical packet switching in future communication networks*. IEEE Commun. Mag., 39(3):128–135, Mar. 2001.

[33] E. F. Burmeister, J. P. Mack, H. N. Poulsen, M. L. Mašanovic, B. Stamenic, D. J. Blumenthal, and J. E. Bowers. *Photonic integrated circuit optical buffer for packet-switched networks*. Opt. Express, 17(8):6629–6635, Apr. 2009.

[34] L. Liu, R. Kumar, K. Huybrechts, T. Spuesens, G. Roelkens, E. J. Geluk, T. de Vries, P. Regreny, D. Van Thourhout, R. Baets, and G. Morthier. *An ultra-small, low-power, all-optical flip-flop memory on a silicon chip*. In Nature Photonics, volume 4, pages 182–187, 2010.

[35] V. Chan. *Optical flow switching*. In Optial Fiber Commun., pages 1 –3, San Diego, CA, USA, Mar. 2010.

[36] E. Mannie. *RFC 3945 : Generalized Multi-Protocol Label Switching (GMPLS) architecture*. Technical report, Network Working Group, Oct. 2004.

[37] E. Rosen, A. Viswanathan, and R. a. Callon. *RFC 3031: Multiprotocol Label Switching Architecture*. Technical report, Network Working Group, Jan. 2001.

[38] D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, and G. Swallow. *RFC 3209 : RSVP-TE: Extensions to RSVP for LSP tunnels*. Technical report, Network Working Group, Dec. 2001.

[39] D. Katz, K. Kompella, and D. Yeung. *RFC 3630 : Traffic Engineering (TE) Extensions to OSPF version 2*. Technical report, Network Working Group, 2003.

[40] J. Lang. *RFC 4204 :Link Management Protocol (LMP)*. Technical report, Network Working Group, 2005.

3

Design and implementation of a simulation environment for network virtualization

"What happens if a big asteroid hits Earth? Judging from realistic simulations involving a sledge hammer and a common laboratory frog, we can assume it will be pretty bad."

–Dave Barry

De Leenheer, M.; Buysse, J.; Mets, K.; Dhoedt, B. & Develder, C., *Design and Implementation of a Simulation Environment for Network Virtualization*, published in Proceedings of the IEEE 16th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), Kyoto, Japan, 10-11 Jun., 2011

This work has been a joint effort between Marc De Leenheer and myself, where my contributions are focused on (i) the design of the complete simulator architecture (see Section 3.4) and (ii) the design and implementation of the NCP+ stack of the simulator.

3.1 Introduction

Current projections indicate that at the end of this decade, the scale of information processing will scale from Petabytes to Exabytes of data [1]. Additionally, emerging paradigms such as cloud computing and IaaS, are driving profound transformations of networks' and users' capabilities [2]. Consequently, a new class of high-performance and high-capacity network-based applications are emerging, posing strict IT (e.g., computing and data storage) resource and service requirements. Due to its own succes and pervasiveness, the current best-effort Internet is unable to adapt to these novel service paradigms. Hence, there is an opportunity for operators/providers to create new services, especially integrated offerings of both optical network connectivity services and traditional IT services.

The GEYSERS project aims to design and showcase a novel architecture, able to provide network operators with an infrastructure composed of several optical network and IT resources in an on-demand fashion [3]. To this end, the physical resources can be partitioned and aggregated to create a virtual infrastructure (VI), which in turn can be controlled by a network operator without interference of other VIs [4]. To control this infrastructure on demand, GEYSERS' architecture deploys an enhanced Network Control Plane (NCP+) that can control both network and IT resources. This way, both network and IT resources can be seen as elements of one homogenous set, able to be provisioned on-demand.

Obviously, validating this architecture is not a straightforward task, given the software and protocol stack that the GEYSERS vision encompasses. As such, the project envisions experiments in a reasonably limited scale testbed comprising around 10–15 nodes. To perform full scale validation, and perform extensive testing of the architecture's scalability, experiments based on discrete event simulations have been identified as the most appropriate method to study the performance. The idea is to implement the full functionality of the layered architecture, and perform validation and testing on medium to large scale networks (consisting of hundreds of nodes). In this paper, we report on the design and implementation of this simulation environment, and demonstrate its features by way of a qualitative discussion of sample simulation scenarios.

The remainder of the paper is organized as follows: Section 3.2 discusses similar proposals, and Section 3.3 introduces the GEYSERS architecture on which our simulation environment is modeled, while Section 3.4 describes the design and some implemention details of the simulator itself. The following Section 3.5 presents a number of use cases that will be validated, and finally Section 3.6 summarizes the paper.

3.2 Related Work

The current interest in architectures for the future internet has led to substantial research on this topic [2]. For instance, sharing a physical infrastructure among multiple virtual networks (only considering networking elements, or, stated differently, disregarding IT end resources), is a topic well-studied and is referred to as Virtual Private Networks (VPN), overlay networks or even active networks.

The goal of a Virtual Private Network or VPN is to connect a number of known end-points over a dedicated communications infrastructure, usually by creating tunnels over a public medium (e.g. the Internet) [5]. These may exist on multiple layers of the network, as evidenced by the existence of either Layer 1, 2 or 3 VPNs. On the other hand, overlay networks are usually implemented on the application layer (L7), and are therefore aimed at providing specific services such as file sharing [6], multicasting [7] or various other goals, including offering Quality of Service (QoS), protecting against Denial of Service (DoS) attacks and many others.

To the best of our knowledge, this paper is the first to report on simulation activities on virtualized networking architectures that comprise combined network+IT virtualisation, and comprise both control plane and a virtualisation layer. Nevertheless, some research has already appeared on the topic of simulation of service-oriented networks. For instance, [8] presents an extensible toolkit for the modelling and simulation of cloud computing environments, while [9] does the same for grid computing infrastructures. Similarly, an example of pure network control plane simulations, esp. GMPLS-based, is [10]. Finally, we mention some work on scalability testing of large networks, as can be found for example in [11, 12], where various approaches are taken to study the performance of different aspects of communication networks.

Complementary to pure simulation studies (as we envisage), also emulation approaches have been proposed to study scalability of large scale networks. For instance, in [13] the authors describe an emulation environment to study fault behaviour and network behaviour in an environment modeled after the Internet.

3.3 The GEYSERS layered architecture

The Generalised Architecture for Dynamic Infrastructure Services (GEYSERS) is a European FP7 project, that has designed a novel architecture for seamless and coordinated provisioning of both optical and IT resources, and developped the necessary tools and software to realize this objective. In particular, virtualization is one of the key goals in this project: adequate mechanisms for abstraction, partitioning and aggregation will be provided. The resources which we consider include optical networking nodes and links, and IT resources such as computational- and

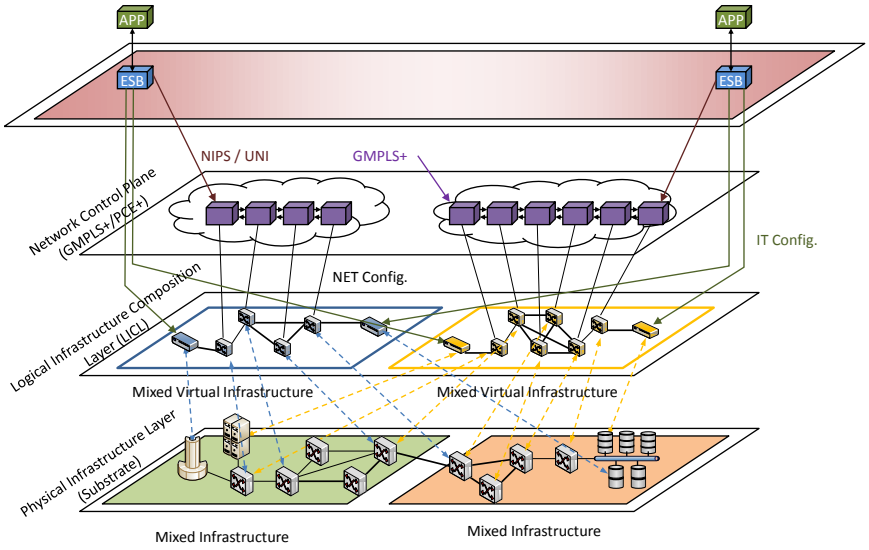


Figure 3.1: GEYSERS layered architecture: the Logical Infrastructure Composition Layer (LICL) offers a framework for abstracting, partitioning and composing virtual infrastructures from a set of physical resources in an automated way. The Network Control Plane (NCP) is a fundamental entity for the seamless provisioning of networked IT services allowing the convergence between IT and network resources.

data storage equipment. Another point of focus is the inclusion of energy efficient mechanisms on all levels of the architecture.

The architecture is detailed in Fig. 3.1, and is basically composed of four layers. First, devices in the Physical Infrastructure (PI) layer are abstracted and partitioned or grouped into virtual resources that can be selected to form the Virtual Infrastructures (VI) in the Logical Infrastructure Composition Layer (LICL). Within each VI, controllers in the IT-aware network control plane (NCP+) layer configure and manage virtual network resources. The Service Middleware Layer (SML) is responsible for translating the application requests and service level agreements (SLAs) into technology specific requests to trigger the provisioning procedures at the NCP+. Refer to [3] for a more detailed discussion on the different components in the layered architecture.

Our aim is to validate the overall GEYSERS architecture, and in particular the end-to-end service provisioning workflow across the various layers and associated interfaces. To this end, we are developing a simulation environment to evaluate the performance of a large-scale network, to complement the relatively small-scale tests of the actual implementation in a testbed. The simulation framework will be used to evaluate the performance and scalability of the architecture and its workflows, as well as associated algorithms for routing, allocation, dynamic

partitioning, etc. Ultimately, the outcome of our research will be used to refine and validate the overall architecture.

3.4 Simulator architecture

We model the major novel components of the aforementioned GEYSERS architecture, specifically the LICL and the NCP+; we do not elaborate on the SML as this component is already in existence in current service-oriented networks.

The main objectives in developing this simulator are to:

- demonstrate the *feasibility* of the GEYSERS architecture (e.g. in terms of achieving energy efficiency)
- identify which potential *bottlenecks* may exist within the architecture
- verify whether the novel components can *scale* towards large networks:
  - comprising a large number of physical resources,
  - supporting a large number of virtual infrastructures,
  - performing as expected under highly dynamic network conditions and user demand.

This requires a thorough investigation of the scalability, overhead, response times and blocking behaviour of the mechanisms, protocols and interfaces.

3.4.1 Overview

The simulator is built on the OMNeT++ simulation framework [14], which is an extensible and highly scalable [15] C++ discrete-event simulation environment aimed at building network simulators. Of particular interest is the INET Framework, which offers implementations for a variety of both wired and wireless networking protocols (covering most of the TCP/IP stack). It also includes an incomplete (but useable) implementation of the (G)MPLS protocol, and forms the basis of our implementation of the NCP+ functionality. The relevant standards include, but not limited to, Resource Reservation Protocol (RSVP), Label Distribution Protocol (LDP), and Contrained Shortest Path First (CSPF) routing.

The simulator is composed of two major blocks (Fig. 3.2): one portion is implemented in OMNeT++, while the other portion makes use of a relational database. This design choice reflects the rather static behaviour of the physical infrastructure, while more dynamic components such as the network control objects are modelled in OMNeT++. This has the advantage of freeing up more memory space for the upper layers of the architecture. Note that as scalability is of major concern, offloading parts of the modeling to a database is preferred, even though a minor penalty can be expected due to database retrieval operations.

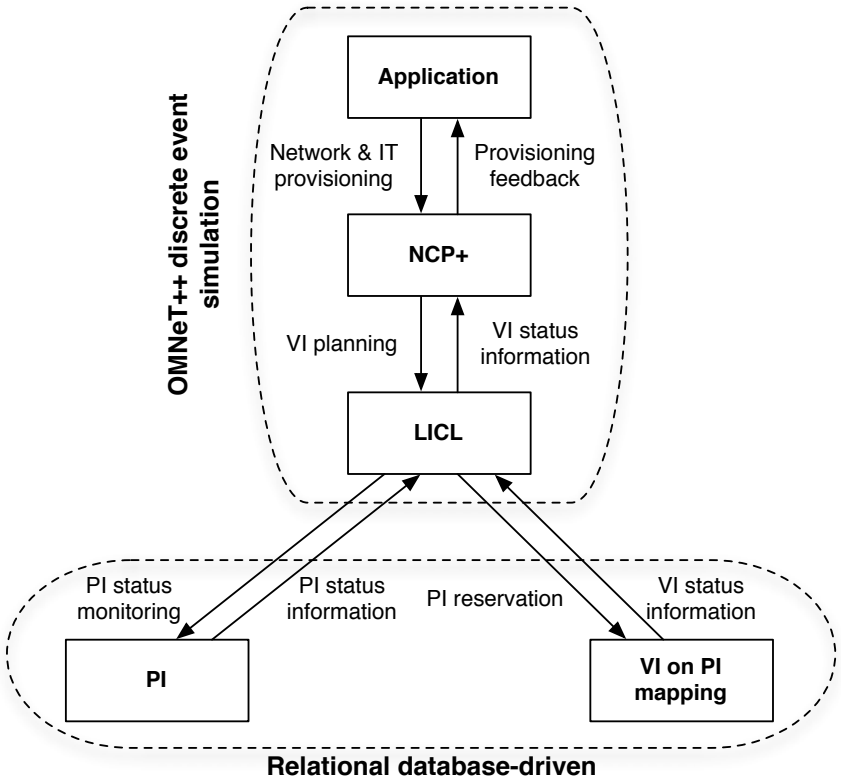


Figure 3.2: Overview of the simulation environment.

3.4.2 UML models

The detailed designs of the different layers introduced are depicted in Fig. 3.3, Fig. 3.4 and Fig. 3.5. The physical infrastructure is composed of devices that correspond to either networking or IT equipment. The key networking entity is the optical cross-connect (`OXC`), which serves as a switching device and is composed of ports (`Port`). Each port can be either an input (`Inport`) or output (`Outport`) port. As we mainly focus on optical networking, each wavelength (`Lambda`) is part of a (`Port`, `Phy Link`) pair, in which the latter represents a physical link. The central object for IT equipment is the physical resource (`Phy Server`), which contains one or more processing units (`CPU`) and storage disks (`Disk`). In its turn, physical resources can be grouped into a cluster or datacenter environment (`Datacenter`). Finally, note the `EnergyController` which provides the energy consumption of various devices (refer to Section 3.5.3 for more detail).

The design of the LICL component (Fig. 3.4) is largely similar to the physical

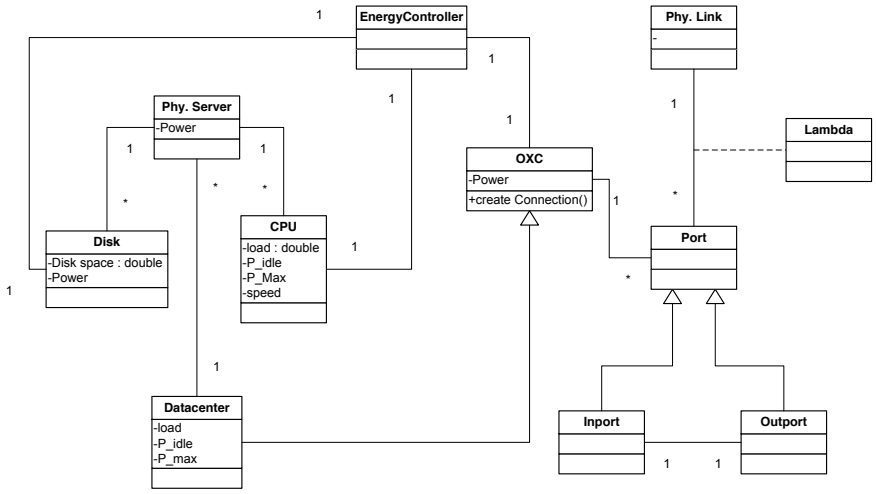


Figure 3.3: Physical infrastructure UML diagram.

infrastructure, since a virtual infrastructure is composed of a partitioning of the physical infrastructure. However, a number of additional classes are necessary, to drive both the planning of the VI process and maintain the mapping between VI and PI. Specifically, the information of the VI-to-PI mappings is stored in the `LICL Resource Inventory`, while the `LICL Partitioning Tool` is responsible for planning. In its turn, the `Planner` can choose between different objectives by selection of an appropriate planning algorithm (`Planning-Algorithm`, see Section 3.5.1).

Finally, the NCP+ simulation component draws from the basic concepts in the GMPLS protocol. A demand for a connection is modeled by the `Request` object, containing the relevant connection parameters. The main element of the network is the `GmplsRouter`, which forms the start and endpoint of an optical connection, represented by a `Route`. This route is calculated by the `PCE+` class, passed on to the `GMPLSRouter`, which then stores this information in the `NodeInfo` object. Each router has a database, consisting of `LinkInfo` objects, to track connections and the wavelengths they use (stored in the `LambdaCap`) for each link. This information is then exchanged through the OSPF-TE and RSVP-TE protocols.

Regarding the IT functionality of the NCP+, the `DataCenter` groups a number of `Server` objects, that are controlled by a `Scheduler`. Finally, the `Message` class is used to exchange information between the different objects in the NCP+.

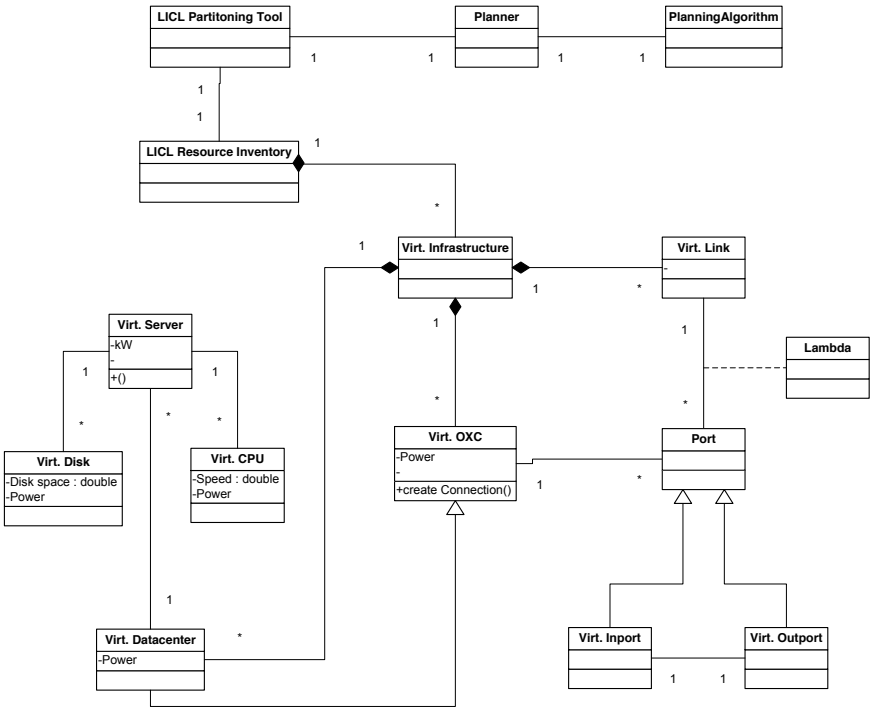


Figure 3.4: LICL UML diagram.

3.4.3 Entity relationship diagram

As shown in Fig. 3.2, a simulation is constructed by running both a discrete event simulator, and a relational database-driven component. In Fig. 3.6, the entity relationship diagram is shown, which contains both the data model for the physical infrastructure and the VI-to-PI mapping. Of note is the detailed description of the optical networking components, as evidenced by the inclusion of transmitter, transceiver, optical switching fabric (MEMS) and the optical amplifiers (EDFA). Also observe the equipment that contributes to the energy consumption of the architecture, in particular the cooling, uninterruptible power supple (UPS) and backup generators, all of which are common in today's communication networks.

3.5 Use cases

In the following, we describe three use cases which will be studied using our simulation environment. These scenarios have served as guidelines during development and are thus the minimal functionalities the simulator supports. The cases are the

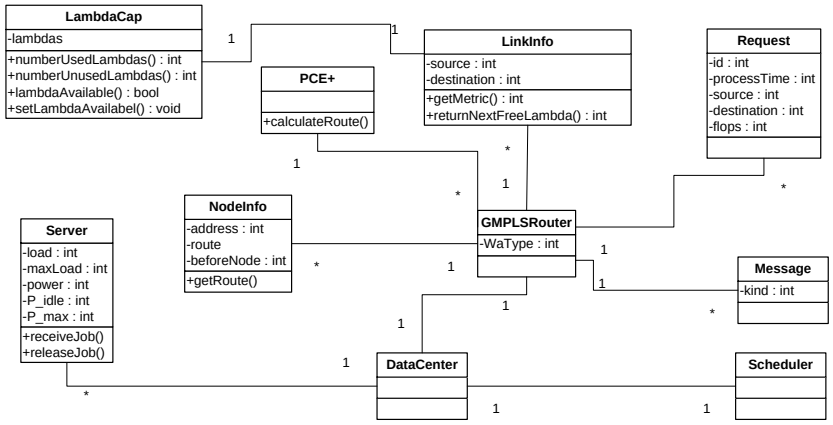


Figure 3.5: NCP+ UML diagram.

result of a consultation of the GEYSERS' partners, comprising representatives from both academia and industry.

3.5.1 LICL Scalability

One of the fundamental issues in virtualized network environments is how to perform the mapping of virtual infrastructure (composed of both network and IT resources) requests on a shared physical infrastructure [16]. A virtual infrastructure (VI) is in essence a subset of the underlying physical infrastructure, and relevant objectives include maximization of the number of accepted VI requests and energy efficient mapping (see Section 3.5.3). This is one of the key roles of the LICL, and thus a number of VI mapping algorithms will be developed and evaluated in the simulation environment.

Referring back to Fig. 3.4, the `Planner` and `PlanningAlgorithm` classes are responsible for this functionality, while the `LICL Partitioning Tool` will make the necessary changes to the `LICL Resource Inventory` based on the VI mapping algorithms' outcome.

Additionally, we will investigate the overhead introduced by the LICL layer, for instance when the NCP+ must provision a new network path, how much delay is added by going through the LICL layer? Finally, an evaluation of the different architectural options for implementing the LICL layer (centralized, distributed, or hybrid) will be performed.

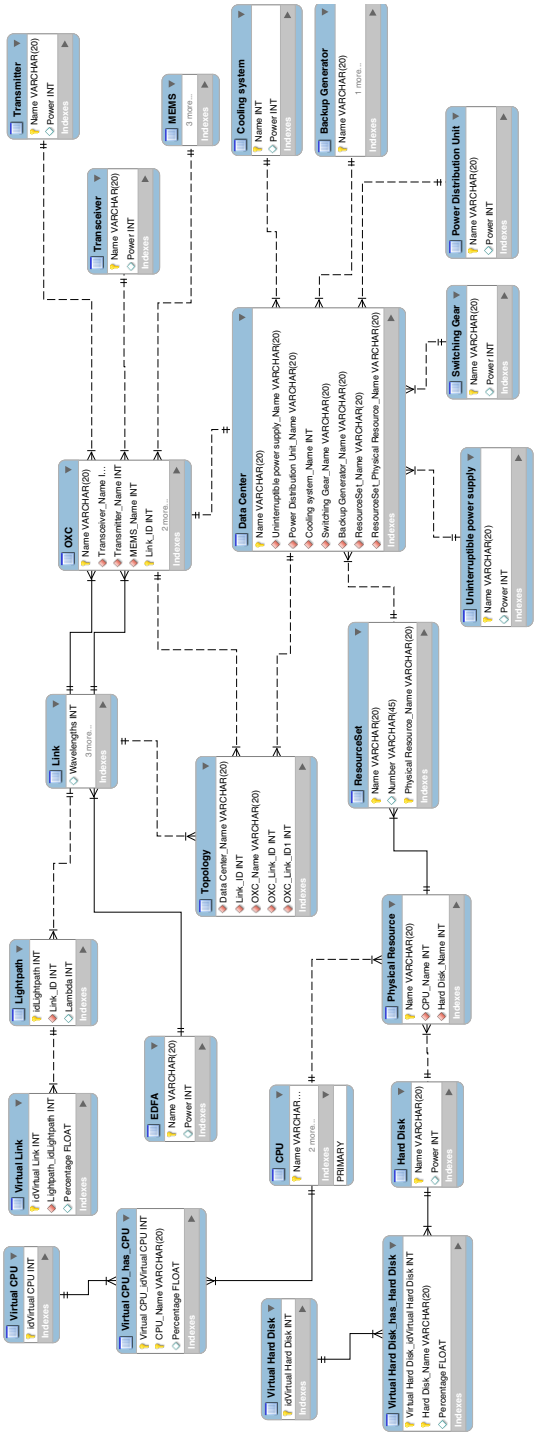


Figure 3.6: Entity relationship model for the physical and virtual infrastructure.

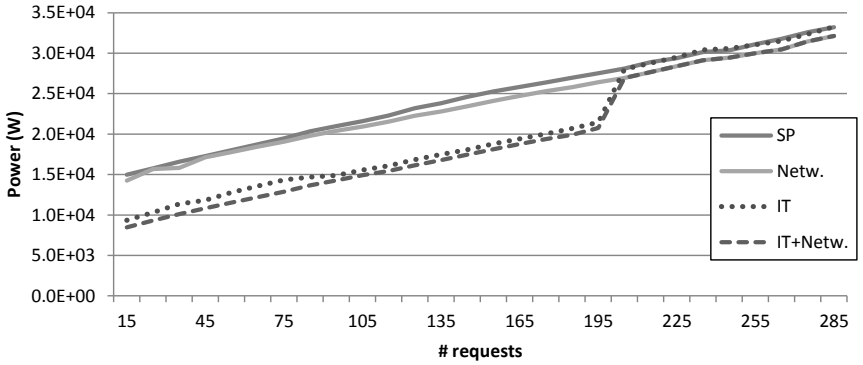


Figure 3.7: Total energy consumption of the COST32 network. The different lines represent the used strategy: SP is shortest path routing, Netw. only considers network resources' power consumption, IT only considers IT resources' power consumption, while IT + Netw. considers both.

3.5.2 NCP+ Scalability

This activity explores the scalability of the proposed NCP+, by evaluating different architectural options for PCE+ inter-domain path computation, and the overhead in terms of, among others, message exchange and signaling delay caused by introduction of IT resource state information in the GMPLS+ protocol. Several strategies are available and will be investigated:

- in the centralized approach, one `PCE+` will do the path computation for all GMPLS+ controllers.
- a hierarchical design, in which a number of GMPLS+ controllers share a `PCE+` object. In turn, these `PCE+` objects share a parent `PCE+`, which performs the path computation on an abstracted topology. A number of alternative aggregation designs will be studied to investigate the scalability of this approach.
- one `PCE+` object per `GmplsRouter`, such that the path computation process is performed in a fully distributed way.

3.5.3 Energy Efficient Design and Operation

The GEYSERS architecture incorporates both energy efficient design and operation of infrastructures, whereby the joint optimization of both IT and optical networking resources is considered [17]. This implies both the LICL, which is reponsible for the VI planning phase (design), and the NCP+, responsible for the

VI service provisioning (operation), should incorporate energy efficiency parameters. As illustrated in Fig. 3.4, these energy-related parameters will be handled by the `EnergyController` in the physical layer. For instance, when two different VI's have virtual resources derived from the same physical resource, the power consumption of each virtual resource is dependent on the total load of the physical resource. Generating the power consumption will be based on both experimental values and appropriate models of the relevant devices in the physical infrastructure (see [17] for examples).

We performed an initial case study (outlined in [17]) to evaluate potential energy savings, by considering both network and IT resource power consumption. Fig. 3.7 shows a number of strategies to reduce energy consumption, and demonstrates that the joint consideration of both network and IT resources can achieve a considerable decrease in energy consumption. Ongoing work involves simulation studies on various online mechanisms to assess the achievable energy savings.

3.6 Conclusion

We presented an overview of the layered GEYSERS architecture, which aims to introduce virtualization of both optical network and IT resources. Furthermore, the design and implementation of a simulation environment, to accurately evaluate the feasibility of the architecture, was presented. The simulator allows extensive scalability testing of all relevant layers of the proposed architecture. Finally, a number of use cases were discussed that demonstrated the functionalities of our simulation environment. A preliminary case study identified substantial potential energy saving opportunities that could be achieved by the Geysers framework.

References

[1] *Cisco white paper - Cisco Visual Networking Index: Forecast and Methodology*. Technical report, CISCO, Jun. 2010.

[2] J. Schonwalder, M. Fouquet, G. Rodosek, and I. Hochstatter. *Future Internet = content + services + management*. IEEE Commun. Magazin, 47(7):27 – 33, 2009.

[3] E. Escalona, P. Shuping, R. Nejabati, D. Simeonidou, J. Garcia-Espin, J. Ferrer, S. Figuerola, G. Landi, N. Ciulli, J. Jimenez, B. Belter, Y. Demechenko, C. De Laat, X. Chen, A. Yukan, S. Soudan, J. Vicat-Blanc, P. ; Buysse, M. De Leenheer, C. Develder, A. Tzanakaki, P. Robinson, M. Brogle, and T. Bohnert. *GEYSERS: A novel architecture for virtualization and co-provisioning of dynamic optical networks and IT services*. In Future Net-

work & Mobile Summit (FutureNetw),, pages 1–8, Warsaw, Poland, 15–17 Jun. 2011.

[4] T. Anderson, L. Peterson, S. Shenker, and J. Turner. *Overcoming the Internet Impasse through Virtualization*. Computer, 38(4):34–41, Apr. 2005.

[5] E. Rosen and Y. a. Rekhter. *RFC 4364 : BGP/MPLS IP Virtual Private Networks (VPNs)*. Technical report, Network Working Group, Feb. 2006. Available from: http://www.ietf.org/rfc/rfc4364.txt.

[6] E. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim. *A survey and comparison of peer-to-peer overlay network schemes*. IEEE Commun. Surveys & Tutorials, 7(2):72–93, 2005.

[7] C. Yang, S. Rao, S. Seshan, and H. Zhang. *Enabling conferencing applications on the internet using an overlay muilticast architecture*. In Proc. 2001 conf. on App., techn., architectures, and protocols for comp. commun., pages 55–67, San Diego, California, USA, 27–31 Aug. 2001.

[8] R. Buyya, R. Ranjan, and R. Calheiros. *Modeling and simulation of scalable Cloud computing environments and the CloudSim toolkit: Challenges and opportunities*. In Proc. Int. Conf. on High Performance Computing & Simulation, pages 1–11, Leipzig, Germany, 21 –24 Jun. 2009.

[9] S. Bagchi. *Simulation of grid computing infrastructure: challenges and solutions*. In Proc. of the Winter Simulation Conference, pages 1–8, Orlando, Florida, U.S.A., 4–7 Dec. 2005.

[10] J. Korniak and P. Roycki. *GMPLS - simulation tools*. In Proc. 1st Conf. "Tools of Information Technology", pages 1–6, Rzeszw, Poland, 15 Sep. 2006.

[11] Y. Garrett R., D. Bauer, H. L. Bhutada, C. D. Carothers, M. Yuksel, and S. Kalyanaraman. *Large-scale network simulation techniques: examples of TCP and OSPF models*. ACM SIGCOMM Computer Communication Review, 33(3):27–41, Jul. 2003.

[12] R. Fujimoto. *Large-scale network simulation: how big? how fast?* In Proc. Int. Symp. on Modelling, Analysis and Simulation of Comp/ Telecommun. Systems, pages 116–123, Washington, DC, USA, 12-15 Oct. 2003.

[13] A. Vahdat, K. Yocum, K. Walsh, P. Mahadevan, D. Kostić, J. Chase, and D. Becker. *Scalability and accuracy in a large-scale network emulator*. In Proc. of the 5th Symp. on Operating Systems Design and Implementation, pages 1–14, Boston, Massachusetts, USA, 9–11 Dec. 2002.

[14] A. Varga and R. Hornig. *An overview of the OMNeT++ simulation environment*. In Proc. of the 1st Int. Conf. on Simu. tools and techniques for commun., networks and systems & workshops, pages 1–10, ICST, Brussels, Belgium, 3–7 Mar. 2008.

[15] E. Weingartner, H. vom Lehn, and K. Wehrle. *A Performance Comparison of Recent Network Simulators*. In IEEE Int. Conf. on Commun., pages 1–5, Dresden, Germany, 14–18 Jun. 2009.

[16] N. M. K. Chowdhury and R. Boutaba. *A survey of network virtualization*. Comput. Netw., 54(5):862–876, Apr. 2010.

[17] A. Tzanakaki, M. Anastasopoulos, K. Georgakilas, J. Buysse, M. De Leenheer, C. Develder, S. Peng, R. Nejabati, E. Escalona, D. Simeonidou, N. Ciulli, G. Landi, M. Brogle, A. Manfredi, E. Lopez, J. Riera, J. Garcia-Espin, P. Donadio, G. Parladori, and J. Jimenez. *Energy Efficiency in integrated IT and optical network infrastructures: The GEYSERS approach*. In Proc. IEEE Conf. on Comp. Commun. Workshops, pages 343–348, Shanghai, China, 10–15 Apr. 2011.

4

Anycast routing for survivable optical grids: scalable solution methods and the impact of relocation

"A common mistake that people make when trying to design something completely foolproof is to underestimate the ingenuity of complete fools."

–Douglas Adams

Shaikh, A.; Buysse, J.; Jaumard, B. & Develder, C., *Anycast routing for survivable optical grids: scalable solution methods and the impact of relocation*, IEEE/OSA Journal of Optical Communication Networks, Vol. 3(9), pp. 767-779, 2011

This work has been a joint effort where my contributions are focussed on (i) proposing the idea of exploiting anycast and relocation to reduce network capacity for shared path protection in optical grids and clouds, (ii) the design, implementation and evaluation of the ILP formulation (see Section 4.3.1), (iii) the design, implementation and evaluation of heuristic H1 (see Section 4.4.1) and (iv) the evaluation of the column generation technique proposed in Section 4.3.2.

4.1 Introduction

Challenging e-Science applications in different domains including high-level computing, parallel programming, fluid-dynamics, astrophysics and climate modeling have given rise to the idea of interconnecting geographically dispersed (high performance) computing sites in so-called grids. A grid is typically defined as a software and hardware infrastructure that provides access to high-end computational resources in a decentralized way, using general-purpose protocols and interfaces while providing a scalable quality of service (QoS) aware architecture [1].

Optical networks, which offer high bandwidth and low latency, are obviously prime candidates for interconnecting the various grid sites. This has given rise to the concept of so-called optical grids [2]. Given the typically high volume of data being processed, it is crucial for grids to be able to survive grid resource failures by providing resiliency mechanisms [2]. This holds for both optical networks and servers (storage and/or computational resources).

In this paper, we consider an optical circuit-switched network (such as an Automatically Switched Optical Network - ASON, see, e.g., [3]), based on Wavelength Division Multiplexing (WDM). We investigate on providing resiliency against network failures with the adoption of *shared path (SP) protection* under the anycast routing principle [4], for the grid to survive from any possible single link failure. We consider two variants of the SP protection schemes for which we provide a generic large scale optimization model, that we compare with the two proposed heuristics and a previously proposed classical integer linear program (ILP), which we adapted to the studied protection schemes.

In Classical Shared Path protection (CSP), a primary path connecting a given pair of source and destination nodes is protected by a link-disjoint backup path. The sharing offers the opportunity to limit the spare network resources, by allowing backup paths to reuse the same physical resources in case the corresponding primary paths are link disjoint. Note that such a scheme can be easily extended to node protection, by requiring node-disjoint paths, instead of link-disjoint paths.

In the context of the usual traffic in a WDM optical network, the node destination is given at the outset together with the description of the traffic requests to be provisioned. However, under the anycast principle [4], which is typical of grids, the destination is not necessarily given a priori. Hence, we will only assume the knowledge of the origin of the grid jobs and let the routing problem decide on an optimized choice of their destination (server location) site. The anycast routing principle even allows identifying a backup-site different from the one under failure-free conditions. This means that, instead of reserving a back-up path to the original destination determined by the grid scheduler, it could be of interest to relocate the job to another server in case of a link failure, assuming tools for seamless transfer of running jobs. As demonstrated further in this paper, exploiting

job relocations allows an overall reduction of (backup) network capacity and can be achieved by a Shared Path Protection with Relocation (SPR-A) scheme under Anycast routing principle. Hence, we will compare two cases, the first one when the backup location is identical to the primary server location (CSP-A scheme), and the second case where freedom is given to select a backup location which may be different from the primary one (SPR-A scheme).

In previous work, Buysse *et al.* [5, 6] demonstrated that such a relocation strategy can significantly decrease the number of network resources (number of bandwidth units) compared to its traditional counterpart, under the assumption of a fully specified traffic matrix, i.e. both job source and destination server nodes were known; relocation was then allowed to change the destination under link failures. Subsequently, in [7], the authors proposed both an ILP model and a heuristic for solving the relocation problem in the anycast case, while optimizing the selection of the destination server site for both working and backup paths.

The current paper significantly extends the aforementioned work with the following contributions: (i) a highly scalable column generation (CG) ILP model and solution; (ii) two new heuristics; (iii) an extensive comparison of the CG-ILP algorithm and of the heuristics, in terms of running times and optimality gaps; (iv) an investigation of the impact of the number of resources (server sites) where to execute the jobs; and (v) an assessment of the bandwidth relocation gains (compared to classical shared path protection) for varying topologies, in terms of average node degree (dense vs. sparse networks).

The remainder of this paper is structured as follows. In Section 4.2, we give an overview of related work. In Section 4.3, we detail the ILP models (previous and new column generation ones), and their solutions. Heuristics are discussed in Section 4.4. The comparative performances of the exact vs. heuristic solutions is dealt with in Section 4.5. In the same section, throughout a case study, we also present the advantage of using relocation, as compared to classical shared path protection. Conclusions are drawn in Section 4.6.

4.2 Related Work

The problem addressed in this paper is a generalization of the classical Routing and Wavelength Assignment (RWA) problem in WDM networks. The vast research literature devoted to RWA focuses on finding a suitable routing path and wavelength, assuming both source and destination of connection requests are given (i.e., the *unicast routing* case). The most studied objectives are the minimum number of wavelengths (min-RWA) and the maximum grade of service, i.e., number of granted requests (max-RWA). For an extensive overview of such classical RWA literature, we refer to [8, 9] and more specifically to the Integer Linear Programming (ILP) approaches reviewed recently in [10–12].

As highlighted before, in this paper, we address the *anycast routing* case, where the problem is complicated by the fact that the destination is not known a priori, but can be freely chosen (among a given set of possible destinations, i.e., server sites). We consider the objective of minimizing the number of wavelengths summed over all network links, i.e., the number of bandwidth units. We consider here an off-line network design problem, aiming to decide on the network and server resource dimensions. Note that we will assume a given set of server sites as destinations; to select them, the approach discussed in [13] can be used. The related problem of accepting arriving connection requests in an on-line fashion (on a given, capacitated network instance), such as considered in [14], is out of the scope of this paper.

ILP formulations have been widely exploited in previous works in order to solve the RWA problem, as they provide a convenient way to flexibly and unambiguously define the problem and its instance-specific parameters: cost functions, wavelength conversion, protection scheme, etc. These ILP formulations typically fall into one of the following two categories: link or path based formulations, see, e.g., [10, 12] for a comparison of them. While some of these ILP formulations are more efficient than others, they all lack scalability when it comes to solving large instances, whether it consists of larger networks or larger traffic data sets. In order to overcome the scalability issues, large scale optimization models need to be devised such as the column generation model of [11]. Therein, the RWA problem is decomposed according to a set of configurations, where a configuration is added only if it contributes to the improvement of the current value of the objective.

While the works described above only have relevance for primary network resource provisioning, their formulations typically only require a few modifications to cover network protection cases. The works of Stidsen *et al.* [15] and Koster *et al.* [16] provide joint optimization of working and protection paths with the classical path protection scheme.

4.3 Proposed solution approaches

We aim to investigate two protection schemes, Classical Shared Path protection with Anycast (CSP-A) and Shared Path protection with Relocation and Anycast (SPR-A) from a network dimensioning perspective, i.e., we extend one step further the Classical Shared Path (CSP) and Shared Path with Relocation (SPR) models which were studied in [17].

We start from a demand vector expressing for every source of an optical grid network, the number of desired connections (i.e., job requests). It is up to the optimization model to choose which primary and backup server sites to use. For the CSP-A protection model, we impose the primary and the backup servers to be the same, while they can differ in the SPR-A model. Furthermore, we assume that ev-

ery optical cross-connect (OXC) in the network is able to perform full wavelength conversion, which is sometimes referred to as the Virtual Wavelength Path (VWP) network [18]. Our network is modeled as follows:

G $= (V, L)$, directed graph representing an optical grid, where V is the node set and L is the set of (directed) links, where we assume that every link has an unlimited transport capacity.

V Node set, indexed by $v \in V$, representing the OXCs and possibly collocated server sites (computational and/or storage servers).

V_d $\subset V$. Server node set, indexed by v or v_d, comprising the server sites (capable of processing grid jobs), i.e., potential candidate destinations.

L Directional link set, indexed by ℓ. Each pair of connected nodes is usually connected by two links, one in each direction.

4.3.1 Standard ILP model

For evaluation purposes, we briefly recall a first standard ILP model which was previously proposed in [6, 7]. We have simplified the notations of its first formulation and adapted it to the protection schemes studied in this paper, i.e., to the CSP-A and SPR-A protection schemes. Note that the first ILP was proposed to study the CSP scheme in which destination server nodes were given at the outset. Traffic instances are described by a set of requests, $k \in K$, where each request k originates at source node $v_s(k)$.

Variables of the first standard ILP model are as follows.

$p^{\mathrm{W}}_{k\ell}$ $\in \{0, 1\}$. $p^{\mathrm{W}}_{k\ell}$ is equal to 1 if request k is routed (working path) through ℓ, 0 otherwise.

$p^{\mathrm{B}}_{k\ell}$ $\in \{0, 1\}$. $p^{\mathrm{B}}_{k\ell}$ is equal to 1 if request k is routed (backup path) through ℓ, 0 otherwise.

d^{W}_{kv} $\in \{0, 1\}$. d^{W}_{kv} is equal to 1 if server site v is used as the primary server site for connection k. (Note that $d^{\mathrm{W}}_{kv} = 0$ for $v \in V \setminus V_d$).

d^{B}_{kv} $\in \{0, 1\}$. d^{B}_{kv} is equal to 1 if server site v is used as the backup server site for connection k. (Note that $d^{\mathrm{B}}_{kv} = 0$ for $v \in V \setminus V_d$).

b^{B}_{ℓ} $\in \mathbb{Z}^+$. b^{B}_{ℓ} is equal to the number of shared backup bandwidth units on link ℓ.

$\delta_{k\ell\ell'}$ $\in \{0, 1\}$. $\delta_{k\ell\ell'}$ is equal to 1 if and only if link ℓ' is used to protect link ℓ on the primary path of connection k.

The objective function aims at minimizing the overall network capacity, in terms of required working and backup bandwidth units on all links:

$$\min \quad \text{COST}_{\text{ILP}}(p^{\text{W}}, b^{\text{B}})$$

where

$$\text{COST}_{\text{ILP}}(p^{\text{W}}, b^{\text{B}}) = \sum_{\ell \in L} \left(b^{\text{B}}_{\ell} + \sum_{k \in K} p^{\text{W}}_{k\ell} \right). \tag{4.1}$$

We next describe the set of constraints. The first set of constraints defines the demand constraints and the flow conservation constraints for the primary paths (where $\omega^{+}(v)$ is the set of v's incoming links, and $\omega^{-}(v)$ that of its outgoing links):

$$\sum_{\ell \in \omega^{+}(v)} p^{\text{W}}_{k\ell} - \sum_{\ell \in \omega^{-}(v)} p^{\text{W}}_{k\ell} = \begin{cases} -1 & \text{if } v = v_k \\ d^{\text{W}}_{kv} & \text{if } v \in V_d \\ 0 & \text{otherwise} \end{cases}$$
$$v \in V, k \in K. \tag{4.2}$$

The next set of constraints expresses the demand constraints and flow conservation constraints for the backup paths:

$$\sum_{\ell \in \omega^{+}(v)} p^{\text{B}}_{k\ell} - \sum_{\ell \in \omega^{-}(v)} p^{\text{B}}_{k\ell} = \begin{cases} -1 & \text{if } v = v_k \\ d^{\text{B}}_{kv} & \text{if } v \in V_d \\ 0 & \text{otherwise} \end{cases}$$
$$v \in V, k \in K. \tag{4.3}$$

Then, we must ensure that working and backup paths do not overlap and do not share any link that could fail simultaneously. For that purpose, we introduce the following constraints:

$$p^{\text{W}}_{k\ell} + p^{\text{B}}_{k\ell} \leq 1 \qquad \ell \in L, k \in K \tag{4.4}$$
$$p^{\text{W}}_{k\ell} + p^{\text{B}}_{k\ell'} \leq 1 \qquad \ell, \ell' \in L : \ell \text{ and } \ell' \text{ are opposite to each other}, k \in K. \tag{4.5}$$

Next, we calculate the shared path protection capacities on each link ℓ:

$$b^{\text{B}}_{\ell} \geq \sum_{k \in K} \delta_{k\ell\ell'} \qquad \ell, \ell' \in L : \ell \neq \ell' \tag{4.6}$$
$$\delta_{k\ell\ell'} \geq p^{\text{W}}_{k\ell} + p^{\text{B}}_{k\ell'} - 1 \qquad k \in K;\ \ell, \ell' \in L : \ell \neq \ell'. \tag{4.7}$$

In order to ensure that every demand (i.e., job request) is assigned to a single server, both in working provisioning and for backup purposes, we enforce the following constraints:

$$\sum_{v_d \in V_d} d_{kv}^{\mathrm{W}} = 1 \qquad k \in K \tag{4.8}$$

$$\sum_{v_d \in V_d} d_{kv}^{\mathrm{B}} = 1 \qquad k \in K. \tag{4.9}$$

Constraints (4.2)-(4.9) define the formulation for the SPR-A protection scheme. In order to get the formulation for the CSP-A scheme, we need to add the following constraints stating that the primary and backup servers have to be the same:

$$d_{kv}^{\mathrm{B}} = d_{kv}^{\mathrm{W}} \qquad v \in V_d, k \in K. \tag{4.10}$$

4.3.2 Column generation ILP model

While column generation techniques allow the solution of very large, even huge, ILP models, they often require to rethink the modeling in order to exhibit a decomposition of the set of constraints, and consequently to allow an implicit enumeration of the variables, its key feature for overcoming non scalability.

Here, in order to get a column generation formulation, we introduce the concept of a configuration $c \in C$, where C denotes the overall set of configurations. A configuration c is defined for a given source node $v_s \in V$, and describes a potential provisioning of the working and backup paths of a set of job requests originating at v_s. In the CSP-A protection scheme, destinations of a pair made of a working and a backup path must be the same server nodes, while in the SPR-A scheme, there is no such requirement. Of course, several such configurations exist and we denote by C_s the set of potential configurations associated with job requests originating at v_s.

The provisioning model of all job requests is then decomposed into: *(i)* a so-called Master Problem (MP) which will select the most promising / best configurations, a sufficient large number so as to satisfy the set of job requests for each source node, and *(ii)* so-called Pricing Problems (PP). Each pricing problem is associated with a given source node and generates potential configurations related to that source node.

The second change we introduce in order to get an efficient column generation is to define the traffic in a slightly different, but equivalent way. Let

K_s Set of job requests originating at source node $v_s \in V \setminus V_d$.

D_s $= |K_s|$, i.e., number of job requests in K_s.

$S \subseteq V$, set of demand source nodes such that:

$$\forall v_s \in S : D_s > 0.$$

To complete the characterization of the configurations, we need the following parameters:

$p^{\mathrm{W}}_{c\ell}$ = 1 if link ℓ is used by the working path of configuration c, 0 otherwise.

$p^{\mathrm{B}}_{c\ell}$ = 1 if link ℓ is used by the backup path of c, 0 otherwise.

The master problem of the column generation ILP model uses two sets of variables: variables $z_c \in \mathbb{Z}^+$, $c \in C$ and $b_\ell \in \mathbb{Z}^+$. The value of each variable z_c is equal to the number of selected copies of configuration c. Variable b^{B}_ℓ is defined as in the ILP model of Section 4.3.1.

4.3.2.1 Master Problem

The objective function which minimizes the total network capacity, can be written as follows:

$$\min \quad \mathrm{COST}_{\mathrm{CG\_ILP}}(z, b^{\mathrm{B}})$$

where

$$\mathrm{COST}_{\mathrm{CG\_ILP}}(z, b^{\mathrm{B}}) = \sum_{\ell \in L} \left(b^{\mathrm{B}}_\ell + \sum_{c \in C} p^{\mathrm{W}}_{c\ell}\, z_c \right). \tag{4.11}$$

The set of constraints are as follows. Firstly, we have the demand (job requests) constraints:

$$\sum_{c \in C_s} z_c \geq D_s \qquad v_s \in S. \tag{4.12}$$

Note that the demand of requests originating at v_s is not necessarily satisfied by a single configuration.

The next set of constraints expresses the capacity requirement for link ℓ' in a backup path. Indeed, if ℓ' protects link ℓ, with ℓ belonging to several working paths (modeled here throughout the various configurations associated with working paths containing ℓ), we must ensure that ℓ' has a large enough transport capacity:

$$\sum_{c \in C} p^{\mathrm{W}}_{c\ell}\, p^{\mathrm{B}}_{c\ell'}\, z_c \leq b^{\mathrm{B}}_{\ell'} \qquad \ell, \ell' \in L : \ell \neq \ell'. \tag{4.13}$$

Note that, in practice, one works with the so-called *restricted master problem*, i.e., with a master problem restricted to a very small set of configuration variables. See Section 4.3.2.4 for a description of the algorithm for solving the CG-ILP model.

4.3.2.2 Pricing Problem

Each pricing problem corresponds to the design of a potential configuration, i.e., a potential working and backup provisioning for the job requests originating from a given source node $v_s \in V$. Per definition of the pricing problem, the objective function corresponds to the reduced cost of the configuration variable of the master problem, i.e., of variable z_c for $c \in C_s$, assuming we search for configurations in C_s. Readers not familiar with linear programming concepts, are referred to [19, 20].

In addition, the interest of the pricing problem lies in the identification of improving configurations, i.e., configurations c such that, if their corresponding variable z_c is added in the master problem, it will contribute to improve (here, to minimize further) the current value of the objective of the master problem. Such configurations are the ones with a negative reduced cost. In other words, assuming we minimize the reduced cost of the current pricing problem associated with source node v_s, either the minimum reduced cost is negative, and then we have obtained an improving configuration that we add to the current master problem, or the minimum reduced cost is positive. In the latter case, we conclude that, at this stage, no more improving configuration associated with v_s can be found, unless the values of the dual variables change following the addition of another configuration associated with another source node.

Let us express the objective function of the pricing problem associated with source node v_s, or PP(v_s) for short, i.e., the reduced cost of variable $z_c, c \in C_s$. For doing so, we need the dual values of the constraints involving variable z_c:

$u^1 \geq 0$, value of the dual vector associated with constraint (4.12-v_s) (we omit the s index to alleviate the notation),

$u^2_{\ell\ell'} \leq 0$, values of the dual vector associated with constraints (4.13).

The reduced cost, $\overline{\text{COST}}_{\text{CG\_ILP}}$, of PP($v_s$), to be minimized, can then be written:

$$\overline{\text{COST}}_{\text{CG\_ILP}} = \sum_{\ell \in L} p^{\text{W}}_{\ell} - u^1 - \sum_{\ell \in L} \sum_{\ell' \in L : \ell \neq \ell'} u^2_{\ell\ell'} p^{\text{W}}_{\ell} p^{\text{B}}_{\ell'}. \tag{4.14}$$

Constraints are related to the working and backup provisioning of the job requests originating from v_s. The next two sets of constraints take care of the work-

ing and backup path definitions.

$$\sum_{\ell\in\omega^+(v)} p_\ell^{\mathrm{W}} - \sum_{\ell\in\omega^-(v)} p_\ell^{\mathrm{W}} = \begin{cases} -1 & \text{if } v = v_s \\ d_v^{\mathrm{W}} & \text{if } v \in V_d \quad v \in V, \\ 0 & \text{otherwise} \end{cases} \tag{4.15}$$

$$\sum_{\ell\in\omega^+(v)} p_\ell^{\mathrm{B}} - \sum_{\ell\in\omega^-(v)} p_\ell^{\mathrm{B}} = \begin{cases} -1 & \text{if } v = v_s \\ d_v^{\mathrm{B}} & \text{if } v \in V_d \quad v \in V. \\ 0 & \text{otherwise} \end{cases} \tag{4.16}$$

The next two sets of constraints deal with the overlap and the sharing of links pertaining to the working and backup paths, and are similar to constraints (4.4) and (4.5):

$$p_\ell^{\mathrm{W}} + p_\ell^{\mathrm{B}} \leq 1 \qquad \ell \in L \tag{4.17}$$

$$p_\ell^{\mathrm{W}} + p_{\ell'}^{\mathrm{B}} \leq 1 \qquad \ell, \ell' \in L : \ell \text{ and } \ell' \text{ are opposite to each other.} \tag{4.18}$$

Again, we need to impose a single node server for each path, i.e., working and backup:

$$\sum_{v\in V_d} d_v^{\mathrm{W}} = 1, \tag{4.19}$$

$$\sum_{v\in V_d} d_v^{\mathrm{B}} = 1. \tag{4.20}$$

This concludes the description of the set of constraints for the SPR-A scheme. For the CSP-A scheme, we have to enforce the constraints stating that the primary and backup servers need to be the same:

$$d_v^{\mathrm{B}} = d_v^{\mathrm{W}} \qquad v \in V_d. \tag{4.21}$$

4.3.2.3 Linearization

As can be observed, the expression of the reduced cost (4.14) is nonlinear. In order to linearize it, we introduce the variables $p_{\ell\ell'}^{\mathrm{WB}}$ such that:

$$p_{\ell\ell'}^{\mathrm{WB}} = p_\ell^{\mathrm{W}} p_{\ell'}^{\mathrm{B}} \qquad p_\ell^{\mathrm{W}}, p_{\ell'}^{\mathrm{B}} \in \{0,1\}; \ell, \ell' \in L : \ell \neq \ell' \tag{4.22}$$

together with the following set of constraints:

$$p_{\ell\ell'}^{\mathrm{WB}} \geq p_\ell^{\mathrm{W}} + p_{\ell'}^{\mathrm{B}} - 1 \qquad \ell, \ell' \in L : \ell \neq \ell' \tag{4.23}$$

$$p_\ell^{\mathrm{W}} \leq p_{\ell\ell'}^{\mathrm{WB}} \qquad \ell, \ell' \in L : \ell \neq \ell' \tag{4.24}$$

$$p_{\ell'}^{\mathrm{B}} \leq p_{\ell\ell'}^{\mathrm{WB}} \qquad \ell, \ell' \in L : \ell \neq \ell'. \tag{4.25}$$

Not that inequalities (4.24) and (4.25) are not necessary, taking into account that variables $p^{\text{WB}}_{\ell\ell'}$ appear in the objective of the pricing problem with a negative coefficient, as $u^2_{\ell\ell'} \leq 0$, (see below) and, hence are minimized.

The expression of the objective (i.e., reduced cost) of the pricing problem PP(v_s) becomes:

$$\overline{\text{COST}}_{\text{CG\_ILP}} = \sum_{\ell \in L} p^{\text{W}}_{\ell} - u^1 - \sum_{\ell \in L} \sum_{\ell' \in L: \ell \neq \ell'} u^2_{\ell\ell'}\, p^{\text{WB}}_{\ell\ell'}. \tag{4.26}$$

4.3.2.4 Solution of the CG-ILP formulation

Column Generation (CG) techniques offer highly efficient solution methods for linear programs with a very large number of variables, where the constraints can be expressed implicitly. In order to speed-up the convergence of a column generation model, it is very often useful to use a "warm" start, i.e., to generate few as promising as possible configurations at the outset. This was achieved by solving PP(v_s) for $v_s \in S$, after modifying its objective as follows:

$$\min \quad \sum_{\ell \in L} \left(p^{\text{W}}_{\ell} + p^{\text{B}}_{\ell}\right). \tag{4.27}$$

The set of constraints is made of constraints (4.15)-(4.21).

On the other hand, one needs to devise a way to derive an integer solution once the linear relaxation of an ILP model has been solved using a column generation algorithm. Here, rather than developing a costly branch-and-cut algorithm, we solve the ILP model made of the columns generated in order to obtain the optimal linear programming solution. It is well known that it usually does not provide the optimal ILP solution, but, as will be seen in the numerical results section, in practice, that was enough in order to obtain near optimal solutions.

The detail of the column generation and ILP solution process is described in Algorithm 1.

Algorithm 1 Solution of the CG-ILP model

Step 1. *Initialization*
Build a set of initial configurations in order to set an initial Restricted Master Problem (RMP).

Step 2. *Solution of the linear relaxation of the master problem*
Solve the LP relaxation of the current RMP
OPT $\leftarrow$ ..FALSE.
while OPT $=$ FALSE **do**
  OPT $\leftarrow$ .TRUE.
  for each source node v_s **do**
    Solve PP(v_s)
    if $\overline{\text{COST}}_{\text{CG\_ILP}}(\text{PP}(v_s)) \leq 0$ **then**
      OPT $\leftarrow$ ..FALSE.
      Add the improving configuration associated with PP(v_s) to the current RMP
      Re-optimize the LP relaxation of the enlarged RMP
    end if
  end for
end while

Step 3. *Deriving an optimal or a near optimal integer solution*
Solve the ILP model made of the current set of columns (variables) of the RMP, using either a branch-and-bound technique or a rounding off technique.

4.4 Heuristics

While the classical ILP formulation presented in 4.3.1 allows to find an optimal solution, it does not scale at all for large data instances. Hence, in order to evaluate the relocation strategy on a larger scale, we proposed, in Section 4.3.2, a new CG-ILP model based on a column generation formulation. This last model allows the solution of large size instances, while providing an optimal or a near optimal solution. We next propose two heuristic algorithms, in an attempt to find faster solution algorithms, without compromising too much on the quality of the solutions. The first heuristic, denoted by H1, improves the running time for medium size instances over CG-ILP, while finding solutions with a small optimality gap. The second one, denoted by H2, is faster than H1, and much faster than the CG-ILP algorithm, but outputs solutions with a larger optimality gap, especially for the CSP-A case. We next describe those two heuristics.

4.4.1 Heuristic H1

We adapted a heuristic described in [21] which tries to minimize the total resource usage by minimizing the resources for the primary connections as well as by maximizing the sharing among the backup network resources. We extended this heuristic to the grid case under the anycast principle, with the selection of the server nodes. We first describe the heuristic for the SPR-A scheme and further show how we can adapt it to achieve the CSP-A scheme.

4.4.1.1 Overview of heuristic H1

Heuristic H1, which is described in Algorithm 2, proceeds in three steps. We next comment those steps.

Step 1: We insert a *virtual resource* (i.e., a sink node) (lines 7-9), which is biconnected with a virtual edge (i.e., two links opposite to each other) of weight 0 to every other resource. Such a virtual resource makes it easy to find a pair of link disjoint paths to different potential resources. If we find two link disjoint paths to this virtual resource, the real resource is the second-last node (next-to-last hop) on each path.

Step 2: For every connection request $k \in K$ (line 12), find a pair of link disjoint paths from the fixed source to the virtual resource (line 13), using Suurballe's algorithm [22] (see also [23]), a reference algorithm for finding two link disjoint paths of minimum total weight. Assign the shortest path to the primary path, p_k^{W} (line 14), and the other path to the backup path, p_k^{B}, (line 15). We choose the longest as backup, since wavelengths along this path will (hopefully) be shared with others. The wavelengths on primary path links on the other hand need to be exclusively reserved for this particular request.

Step 3: For every connection (line 21), try to find a new primary resource (line 24, using Dijkstra's algorithm [24]). The search of the new backup path (line 25), using the procedure FindBackupPath, is described in algorithm 3. Therein, we first delete the primary path, after which we consider every connection $k' \neq k$. If primary paths $p^{\mathrm{W}}_{k'}$ and p^{W}_{k} are link disjoint, we assign weight 0 to the links $\ell \in p^{\mathrm{B}}_{k'}$. Applying Dijkstra's algorithm on the modified network from the source to the virtual resource leads to a new backup path with a cost hopefully not greater than the cost of the previous backup path and even smaller because of possible additional sharing. This last step differs from [21] as we combine the separate rerouting steps into one step. Such a combination accommodates for the extra degree of freedom (vs. [21]) since we start from a source demand vector, rather than from an origin/destination demand matrix.

In order to accommodate all backup paths, the total number of bandwidth units on each link ℓ is calculated as follows:

$$b^{\mathrm{B}}_{\ell} = \max_{\ell' \in L} \sum_{k \in K} p^{\mathrm{B}}_{k\ell} \cdot p^{\mathrm{W}}_{k\ell'}. \tag{4.28}$$

4.4.1.2 Extending H1 heuristic for the solution of CSP-A

The introduction of the virtual resource is a handy trick in order not to exhaustively optimize over all possible resources and then choosing the best one. The trick cannot be used for CSP-A because the end points of the pair of link disjoint paths need to be the same. Hence, there is no other possibility than exhaustively iterate over every possible resource in both the initial configuration phase and the optimization phase. Note, however, that this exhaustive search for all resources is feasible, since we assume a reasonably small set V_d of resource sites. This choice is motivated by [13] which shows that a small number of resource sites suffices and allows the minimization of the overall network load.

In order to get a solution for the CSP-A protection scheme, heuristic H1 should be modified as follows:

- Remove Step 1 (lines 7 to 9),
- For each server site, calculate a new primary path and an optimized backup path, following the approach of lines 14–15, and choose the combination that leads to the lowest bandwidth requirements (i.e. that minimizes COST_-H1).

Algorithm 2 Heuristic H1 - SPR-A Protection Scheme

Step 0. *Initialization*
for $k \in K$ **do**
    $p_k^{\text{W}} \leftarrow \emptyset$; $p_k^{\text{B}} \leftarrow \emptyset$
end for

Step 1. *Create virtual resource* v_{n+1}
for $v \in V_d$ **do**
    create two parallel links between v and v_{n+1}, where node v_{n+1} plays the role of a sink node.
end for

Step 2. *Find a candidate link disjoint pair of paths*
for $k \in K$ (where K is an ordered set) **do**
    $(p_1, p_2) \leftarrow$ Suurballe's algorithm(s, v_{n+1})
    $p_k^{\text{W}} = \arg\min(\text{LENGTH}(p_1), \text{LENGTH}(p_2))$
    $p_k^{\text{B}} = \arg\max(\text{LENGTH}(p_1), \text{LENGTH}(p_2))$
end for
Compute COST_H1 associated with those primary and backup paths

Step 3. *Optimization phase*
# Changes $\leftarrow 0$; $index \leftarrow -1$
while # Changes $\leq |K|$ **do**
    $index = (index + 1) \mod |K|$
    $k \leftarrow K\,[index]$
    $p_k^{\text{W}} \leftarrow$ Dijkstra's algorithm$(v_s(k), v_{n+1})$
    $p_k^{\text{B}} \leftarrow$ FindBackupPath$(v_s(k), p_k^{\text{W}}, v_{n+1})$
    Compute NEW_COST_H1
    if NEW_COST_H1 ¡ COST_H1 **then**
        # Changes $\leftarrow 0$
        COST_H1 $\leftarrow$ NEW_COST_H1
    else
        # Changes $\leftarrow$ # Changes + 1
    end if
end while

Algorithm 3 Algorithm FindBackupPath$(v_d, p_k^{\text{W}}, v_{n+1})$

Remove the links of p_k^{W} in graph G
For each backup path with a corresponding primary that is disjoint with p_k^W, set the link weights to zero
for $k' \in K \setminus \{k\}$ **do**
    if $(p_k^{\text{W}} \cap p_{k'}^{\text{W}}) = \emptyset$ **then**
        for $\ell \in p_{k'}^{\text{B}}$ **do**
            assign weight 0 to ℓ
        end for
    end if
end forreturn Dijkstra's algorithm$(v_s(k), v_{n+1})$

4.4.2 Heuristic H2

In this section, we describe another heuristic algorithm, H2, in an attempt to design a more scalable heuristic algorithm than heuristic H1. As we will see in Section 4.5, we were quite successful in that attempt for the scalability aspect, less for the accuracy part. A key difference between H1 heuristic and H2 heuristic is that in H2, we combine all the requests originating from the same source node, as in the master problem of CG-ILP, while in H1, requests are considered on an individual basis, which increases the complexity of H1.

The H2 heuristic is based on an iterative approach which is detailed in Algorithm 4.

Shortest paths are computed using different weights for primary and backup path calculation. Backup weights account for sharing of wavelengths, while working weights account for the length of the path only:

WEIGHT$^{\mathrm{W}}_{\ell}$: Primary weights are all taken equal to one, meaning that when computing shortest paths with those weights, we indeed consider the length of the working paths in terms of the number of links they contain.

WEIGHT$^{\mathrm{B}}_{\ell}$: Backup weights are initialized to one, and will contain the complement of the protection bandwidth requirements with respect to the maximum link bandwidth requirement, see line 9. The reason is as follows. When computing shortest paths, we can either minimize or maximize their overall bandwidth requirements. When maximizing, instead of changing the shortest path algorithm in a longest path algorithm, one can also complement the protection weights with respect to the largest weight in order to go on using a shortest path algorithm (this is what is done on line 9 of the heuristic).

The underlying idea of the definition of the weights for the search of the backup path is that there are more opportunities for sharing with the links already contributing to bandwidth protection, or, in other words, the more protection bandwidth a link has, the more protection bandwidth sharing the link offers. For a given source node, there might be several requests. It is the choice of the network manager to route them all on the same primary paths or not. Indeed, it is not mandatory to assign each of the requests originating at the same node with the same server, and to assign them the same backup path. However, this is the choice which has been made in the H2 heuristic for scalability purposes.

4.4.2.1 Extending H2 heuristic for the solution of CSP-A

As for heuristic H1, heuristic H2 can be easily adapted to the CSP-A scheme: the search for paths simplifies as they are restricted to pairs of working/backup paths with the same destinations.

Algorithm 4 Heuristic H2 - SPR-A Protection Scheme (Part 1)

Step 1: *Initialization*
For all $\ell \in L$: $b_\ell^{\mathrm{B}} \leftarrow 0$; $\mathrm{WEIGHT}_\ell^{\mathrm{W}} \leftarrow 1$,

Step 2: *Primary and backup paths*
for all $v_s \in V \setminus V_d$ **do**
Concatenate all the requests originating at v_s into a single aggregated request, denoted by $k(v_s)$, with a bandwidth requirement such that: $b_{k(v_s)} = \sum_{k \in K_s} b_k$.
Step 2a: *Selection of the grid server location*
for all $\ell \in L$ **do**

$$\mathrm{WEIGHT}_\ell^{\mathrm{B}} \leftarrow \left(\max_{\ell \in L} b_\ell^{\mathrm{B}} \right) - b_\ell^{\mathrm{B}} + 1$$

end for
for all $v_d \in V_d$ **do**
Compute the shortest path $p_{v_s v_d}$ from v_s to v_d with weights $\mathrm{WEIGHT}^{\mathrm{W}}$
end for
$p_s^{\mathrm{W}} \leftarrow \arg\min_{v_d \in V_d} \{\mathrm{LENGTH}(p_{v_s v_d})\}$ where $\mathrm{LENGTH}(p_{v_s v_d})$ is computed according to $\mathrm{WEIGHT}^{\mathrm{W}}$

Step 2b: *Tentative selection of the primary path*
Temporarily remove from G the links of p_s^{W}

Step 2c: *Selection of the backup path and confirmation/new computation of the primary path*
if there exists a path from v to a server site **then**
For all $v_d \in V_d$: Compute the shortest path $p_{v_s v_d}$ from v_s to v_d with weights $\mathrm{WEIGHT}^{\mathrm{B}}$
$p_s^{\mathrm{B}} \leftarrow \arg\min_{v_d \in V_d} \{\mathrm{LENGTH}(p_{v_s v_d})\}$ where $\mathrm{LENGTH}(p_{v_s v_d})$ is computed according to $\mathrm{WEIGHT}^{\mathrm{B}}$
Restore graph G (put back all links)
else
Restore initial graph G (put back all links)
Compute the shortest pair of link disjoint paths between v_s and v_d with weights $\mathrm{WEIGHT}^{\mathrm{W}}$ and $\mathrm{WEIGHT}^{\mathrm{B}}$, for all $v_d \in V_d$.
Let p' and p'' be the two resulting routes. Let

$$p_s^{\mathrm{W}} = \arg\min \{ \mathrm{LENGTH}(p'), \mathrm{LENGTH}(p'') \}; \tag{4.29}$$

$$p_s^{\mathrm{B}} = \arg\max \{ \mathrm{LENGTH}(p'), \mathrm{LENGTH}(p'') \}. \tag{4.30}$$

end if

Algorithm 5 Heuristic H2 - SPR-A Protection Scheme (Part 2)

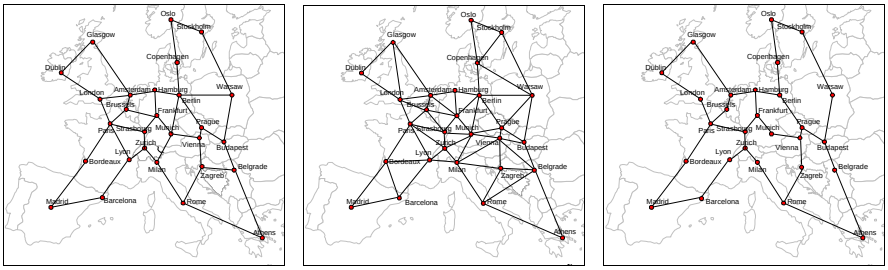
29: Update the bandwidth requirements (b_ℓ^{W} and b_ℓ^{B}) on the links of the primary and backup paths. For b_ℓ^{B}, the updating formula is as follows:

$$b_\ell^{\mathrm{B}} = \max_{\ell' \in L} \left(\sum_{k \in K : \ell' \in p_k^{\mathrm{W}}, \ell \in p_k^{\mathrm{B}}} b_k \right), \tag{4.31}$$

where p_k^{W} (resp. p_k^{B}) is the aggregated working (resp. backup) path of request k.

30: **end for**

4.5 Performance evaluation



(a) EU-base (the base topology from [25])

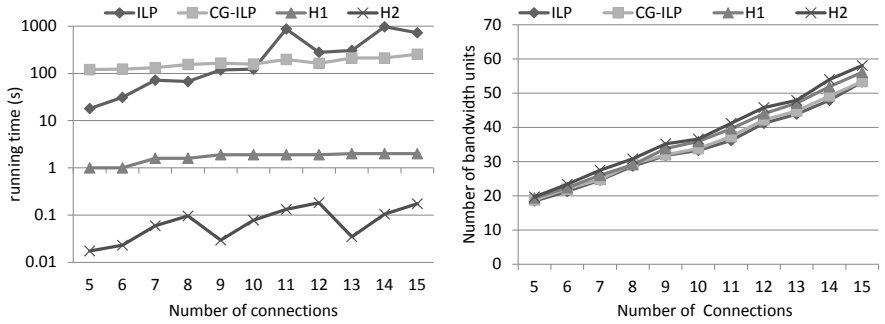
(b) EU-dense (the triangular topology from [25])

(c) EU-sparse (the ring topology from [25])

Figure 4.1: The original pan-European network topology and two variants of it.

4.5.1 Experiment set-up

We will compare the performances for the Classical Shared Path Protection (CSP-A) and the Shared Path Protection with Relocation (SPR-A) schemes, both under the Anycast routing principle. In order to evaluate the influence of the topology on the achievable savings, we will compare three different topologies [25] as depicted in Fig. 4.1: (a) *EU-base*: a meshed network topology comprising 28 sites and 41 bidirectional links, corresponding to the pan-European network of the LION and COST ACTION 266 projects; (b) *EU-dense*: a denser variant, with the same number of nodes, but 59 bidirectional links; and (c) *EU-sparse*: a sparser variant, again with the same node set, but with only 35 bidirectional links.



(a) Computing times (logarithmic scale)

(b) Number of bandwidth units

Figure 4.2: Compared Performances of ILP, CG-ILP, H1 and H2 on small data sets (SPR-A protection scheme).

Traffic instances were generated as follows: for a given number, say $|K|$, of job requests, we randomly select $|K|$ source nodes $v_s \in V \setminus V_d$. The number of times a source node is selected gives the number of job requests originating from that node. Nodes which are hosting server nodes are excluded.

We compare the solutions of the two ILP models, as well as the solutions of the two heuristics described in the previous sections. We consider different sets of fixed server nodes:

$V_d^3 = \{\text{London, Vienna, Berlin}\}$

$V_d^5 = V_d^3 \cup \{\text{Lyon, Zurich}\}$

$V_d^7 = V_d^5 \cup \{\text{Munich, Zagreb}\}$

We use the IBM ILOG CPlex solver (release 11) to solve the ILP models under a C++/java implementation. All programs have been run on a cluster server node with 1 CPU of 2.2 GHz AMD Opteron 64-bit processor, 8Gb ram. In the forthcoming figures, each data point corresponds to average results over 10 random traffic instances.

4.5.2 Quality of the solutions

4.5.2.1 Accuracy of the solutions

Before comparing the performances of the CSP-A and SPR-A protection schemes, it is necessary to have a look at the quality (i.e., accuracy) of the solutions output by the CG-ILP algorithm and the two heuristics. In order to do so, we conducted experiments on the base pan European network topology of Fig. 4.2a, with 5 server nodes (set V_d^5).

In our previous work [17], we already compared the quality of the solutions provided by CG-ILP and an earlier version of H1, noted as H1′, for both the classical shared path protection (CSP) and the shared path protection with relocation (SPR) schemes, assuming the location of the servers was given at the outset, in the description of each job request. Therein, we observed that both CG-ILP and H1′ found very close solutions (less than 1% optimality gap) to the optimal ILP solution, on small instances, i.e., with a number of requests less than 20. On larger instances, the ILP model is not scalable anymore, and we observed that CG-ILP and H1′ solutions were very close, with the CG-ILP algorithm being faster than H1, the more so as the number of requests was increasing. In addition, the optimality gap of CG-ILP was equal to 1% on average, while it was equal to $\approx 5\%$ for heuristic H1′.

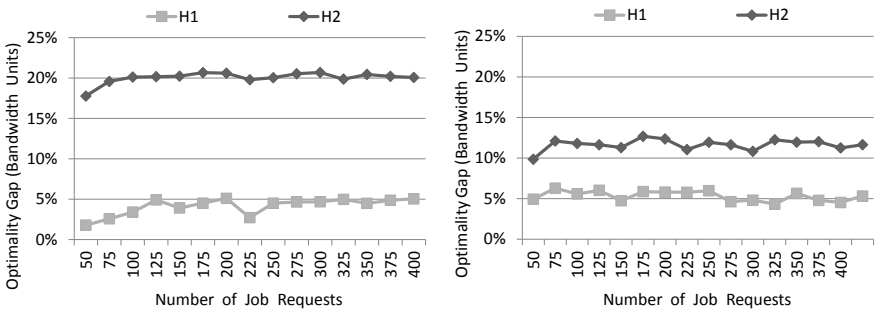
If we now look at the CSP-A and SPR-A protection schemes, where the server location is not given at the outset (in comparison with the CSP and SPR schemes in [17]), we observe similar results for small data sets. We only provide the results for the SPR-A protection scheme, since the qualitative results for the schemes, CSP-A and SPR-A, are very similar. Indeed, for small data sets, where the classical ILP model remains solvable, see Figure 4.2, we observe that the ILP and the CG-ILP solutions are very close, meaning that the CG-ILP model leads to near optimal solutions which are within less than 2% accuracy[1], while the H1 heuristic finds solutions close to the optimal one ($\geq 2\%$ accuracy), see Figure 4.3b. The comparison also includes heuristic H2, which is a faster heuristic than H1, at the expense of a larger optimality gap of 9%. With respect to the computing times (see Figure 4.3a), the heuristics are much faster than the two ILP algorithms. Observe that the ILP model's lack of scalability is visible from the clear increase in running time for larger demands (note the logarithmic scale), whereas the running time for CG-ILP seems more stable for increasing demands.

For larger data sets, the results are described in Figure 4.3. We have noted that CG-ILP has an optimality gap $< 2\%$ which means we get optimal solutions from a practical point of view. In both figures, we provide the relative performances of the two heuristics, H1 and H2, with respect to CG-ILP. The relative optimality gaps are computed as follows:

$$\frac{\text{COST}^{\star}_{\text{H1}} - \text{COST}^{\star}_{\text{CG-ILP}}}{\text{COST}^{\star}_{\text{H1}}} \quad \text{and} \quad \frac{\text{COST}^{\star}_{\text{H2}} - \text{COST}^{\star}_{\text{CG-ILP}}}{\text{COST}^{\star}_{\text{H2}}},$$

where $\text{COST}_{\square}$ denotes the cost value found by the $\square$ model/algorithm. Comparisons are made in Figure 4.4a for the CSP-A protection scheme , and in Figure 4.4b for the SPR-A protection scheme. The key observations are that the H1 heuristic provides better solutions than the H2 heuristic, but at the expense of longer computing times, as discussed below. Indeed, for both protection schemes, the H1

[1] Accuracy is defined by the optimality gap compared to ILP for small instances, and to CG-ILP for large instances.



(a) CSP-A (b) SPR-A

Figure 4.3: Performances of H1 and H2 compared to CG-ILP.

heuristic provides solutions with an average of 5% accuracy, compared to the CG-ILP solutions, while the relative accuracy varies between 10% and 20% for the H2 heuristic.

4.5.2.2 Computing times vs. solution accuracy

The results discussed here have again been obtained for the EU-base topology with 5 server sites. The optimal solution of the ILP model can only be compared with the other solutions on small data sets. There, we observe that the CG-ILP model very quickly provides near optimal solutions with a very good accuracy (less than 2%). In addition, on average, CG-ILP has smaller computing times than ILP as soon as we have more than 10 connection requests. The H1 solution is less accurate, with a consistent gap around 5%, for both the CSP-A and SPR-A schemes, but its computing times are much smaller than those of the solutions for the ILP models. Similar observations can be made for H2, which is even faster than H1, but with a reduced accuracy (around 9%).

On larger data sets, only the solutions of the CG-ILP, H1 and H2 algorithms can be compared, see Figure 4.4. We observe that both CG-ILP and H2 algorithms are not sensitive to the number of requests, with H2 being much faster than CG-ILP. On the other hand, H1 is increasing with the number of requests, and when the number of requests exceeds 500, H1 has higher computing times than CG-ILP. As shown by the results depicted in Figure 4.3, H1 provides better solutions than H2. However, when accuracy is not a major concern, but routes need to be found very fast, H2 is an interesting alternate choice and scales to very large demand sets.

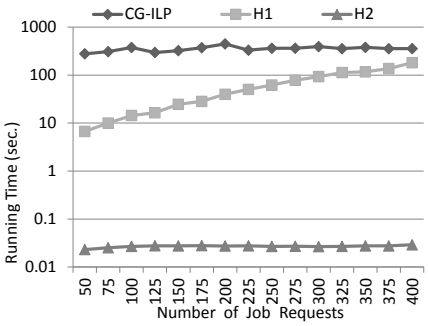


Figure 4.4: Running time for SPA-R protection scheme.

(a) CSP-A (b) SPR-A

Figure 4.5: Comparison of the running times for different numbers of server nodes on the EU-base topology (CG-ILP algorithm).

4.5.3 Influence of the number of server sites and the topology

4.5.3.1 Number of servers

We compare here the performances of the CG-ILP algorithm with different numbers of resources (server nodes): 3, 5, and 7. Results are shown in Figure 4.6a (resp. 4.6b) for the CSP-A (resp. SPR-A) protection scheme. We observe, that for the CSP-A scheme, computing times are higher for 5 server locations than for 3, while computing times for 3 are higher than those for 7 server locations. For the SPR-A scheme, the running times with 3 server nodes are higher than with 5, and running times with 5 server nodes are higher than those with 7 server locations. We made experiments with a different data set, where the Berlin server was relocated in Copenhagen. Again, the results (not shown here) gave similar running times for 3 and 5 server locations, and lower ones for 7 server locations than for 3 or 5 server locations. Therefore, from the two case studies, no clear trend can be

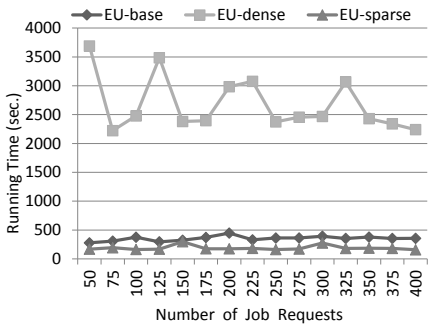


Figure 4.6: Impact of the topology connectivity (CG-ILP algorithm): Running times for the SPR-A protection scheme.

observed in runtime dependency on the number of server sites.

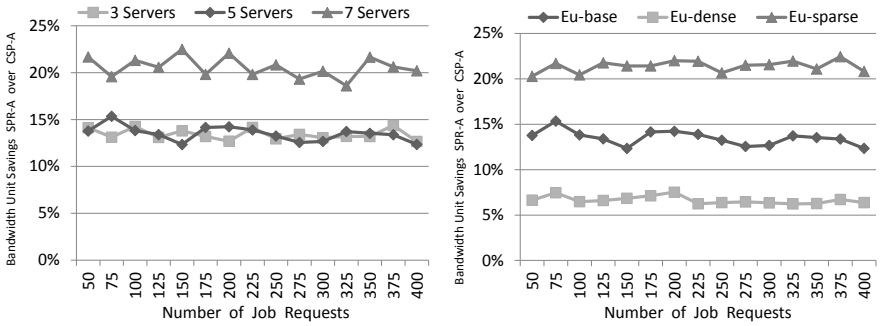
4.5.3.2 Impact of the topology connectivity

We next analyze the effect of the topology. For doing so, we considered the EU networks comprising the same number of nodes, but with different number of links (i.e., connectivity). We again considered the case for 5 server sites. Consequently, we investigate the performance of algorithm CG-ILP on the 3 topologies of the pan-European network (see Figure 4.1) described at the beginning of Section 4.5: EU-base, EU-dense, EU-sparse with an average node degree of 2.5, 4.21, and 2.93 respectively.

Contrarily to the number of server sites, the topology seems a lot more influential, where a highly meshed network severely penalizes the execution time for CG-ILP, as observed in Figure 4.6. This was to be expected, since the number of possible paths increases.

4.5.3.3 Bandwidth savings by exploiting relocation

Lastly, we compared the bandwidth requirements of CSP-A and SPR-A, depending on the number of server nodes and the network topology. In Figure 4.7, we plotted the bandwidth savings that result from using the SPR-A scheme rather than the CSP-A scheme, using the ratio (bandwidth (CSP-A) – bandwidth (SPR-A)) / bandwidth (CSP-A). In all cases, there are meaningful bandwidth savings, which is rather stable with the number of job requests (experiments have been conducted for 50 up to 400 requests). On average, it is around 13% for 3 and 5 servers, and increases to around 21% for 7 servers. Indeed, the more servers, the more flexibility for an anycast scheme. With respect to the impact of the topology, the trend is as expecting, more bandwidth savings as the density is decreasing, i.e., bandwidth savings go from an average of 7% on a dense topology, to an average of 13% for



(a) Depending on the # of server nodes on the EU-base topology

(b) Depending on the density of the network topology with 5 server sites

Figure 4.7: SPR-A vs. CSP-A protection schemes with respect to the number of bandwidth units.

the base topology, and then to above 21% for the sparse topology. It can be explained since, in such a ring-like network, a backup path to the same destination as the primary is likely to be quite long, and quite a bit longer (on average) than a path towards another server site.

4.6 Conclusion and future work

In this paper, we formulated an ILP for network dimensioning purposes in an optical grid scenario with shared path protection against network single link failures. The fundamental difference with traditional RWA problems stems from the anycast routing principle: we also need to decide on the destination of the grid traffic (i.e. which grid server processes the submitted jobs originating from a particular source). Extensive case studies showed that solving the flow formulation ILP is not scalable, hence, we proposed heuristics able to solve large problem instances (with case studies ranging to networks of 28 nodes and 59 bidirectional links, and up to 400 connections). In addition, we also proposed a scalable exact method (CG-ILP) relying on column generation techniques, which offers a small to very small optimality gap (0.15 % and 1.8% on average for CG-ILP on large and small instances respectively).

With respect to the shared path protection scheme, we extended our earlier limited case studies [7] on assessing the amount of network bandwidth savings achievable by exploiting relocation. We investigated the influence of network topology, and in particular node degrees, on potential savings. We found that for lower node degrees, hence sparser networks, the potential savings are much higher; 21% for a European network with 28 nodes and average node degree of 2.5 (Fig. 4.2b) ,

versus 7% for node degree 4.21 (Figure 4.2c).

The network savings of our relocation strategy come at the price of increased load on the relocation servers. However, in reality this seemingly additional cost is one that would need to be made anyhow to provide resilience against server failures. Our future work will investigate this claim in more detail, by studying relocation-based protection mechanisms that offer survivability in case of both single node and single link failures.

References

[1] I. Foster and C. Kesselman. *The grid: blueprint for a new computing infrastructure*. Morgan Kaufmann, San Francisco, CA, USA, 2nd edition, 2004.

[2] M. De Leenheer, C. Develder, T. Stevens, B. Dhoedt, M. Pickavet, and P. Demeester. *Design and control of optical grid networks*. In Proc. 4th Int. Conf. on Broadband Networks (Broadnets), pages 107–115, Raleigh, NC, 10–14 Sep. 2007.

[3] A. Jaiszczyk. *Automatically switched optical networks: benefits and requirements*. IEEE Commun. Mag., 43(2):10–15, Feb. 2005.

[4] T. Stevens, M. De Leenheer, C. Develder, F. De Turck, B. Dhoedt, and P. Demeester. *Anycast routing algorithms for effective job scheduling in optical grids*. In Proc. of European Conf. on Opt. Commun. (ECOC), pages 1–2, Cannes, France, 24–28 Sep. 2006.

[5] J. Buysse, M. De Leenheer, C. Develder, and B. Dhoedt. *Exploiting relocation to reduce network dimensions of resilient optical grids*. In Proc. 7th Int. Workshop Design of Reliable Commun. Netw. (DRCN), pages 100–106, Washington, D.C., USA, 25–28 Oct. 2009.

[6] J. Buysse, M. De Leenheer, B. Dhoedt, and C. Develder. *Providing resiliency for optical grids by exploiting relocation: A dimensioning study based on ILP*. Comput. Commun., 34(12):1389–1398, 2011.

[7] J. Buysse, M. De Leenheer, B. Dhoedt, and C. Develder. *On the impact of relocation on network dimensions in resilient optical grids*. In Proc. 14th Int. Conf. on Optical Network Design and Modelling (ONDM), Kyoto, Japan, 31 Jan.–3 Feb. 2010.

[8] R. Dutta and R. G.N. *A survey of virtual topology design algorithms for wavelength routed optical networks*. Optical Netw. Mag., 1(1):73–89, Jan. 2000.

[9] H. Zang, J. P. Jue, and B. Mukherjee. *A Review of Routing and Wavelength Assignment Approaches for Wavelength-routed Optical WDM Networks*. Optical Netw. Mag., 1:47–60, Jan. 2000.

[10] B. Jaumard, C. Meyer, and B. Thiongane. *Comparison of ILP formulations for the RWA problem*. Optical Switch. and Netw., 4(3):157–172, Nov. 2007.

[11] B. Jaumard, C. Meyer, and B. Thiongane. *On column generation formulations for the RWA problem*. Discrete Applied Mathematics, 157(6):1291–1308, Mar. 2009.

[12] B. Jaumard, C. Meyer, and B. Thiongane. *ILP formulations for the RWA problem for symmetrical systems*. In P. Pardalos and M. Resende, editors, Handbook for Opt. in Telecommun., chapter 23, pages 637–678. Kluwer, 2006.

[13] C. Develder, B. Mukherjee, B. Dhoedt, and P. Demeester. *On dimensioning optical grids and the impact of scheduling*. Photonic Netw. Commun., 17(3):255–265, Jun. 2009.

[14] X. Liu, C. Qiao, W. Wei, X. Yu, T. Wang, W. Hu, W. Guo, and M.-Y. Wu. *Task scheduling and lightpath establishment in optical grids*. Journal of Lightwave Technology, 27(12):1796–1805, Jun. 2009.

[15] T. Stidsen, B. Petersen, S. Spoorendonk, M. Zachariasen, and K. Rasmussen. *Optimal routing with failure-independent path protection*. Networks, 2(55):125–137, Mar. 2010.

[16] A. Koster, A. Zymolka, M. Jger, and R. Hulsermann. *Demand-wise shared protection for meshed optical networks*. Journal of Network and Systems Management, 13(1):35–55, 2005.

[17] B. Jaumard, J. Buysse, A. Shaikh, M. De Leenheer, and C. Develder. *Column generation for dimensioning resilient optical grid networks exploiting relocation*. In Proc. IEEE Global Telecommun. Conf. (Globecom), pages 1–6, Miami, FL, USA, 6–10 Dec. 2010.

[18] K.-I. Sato. *Advances in transport network technologies: photonic networks, ATM and SDH*. Artech House Publishers, 1996.

[19] V. V. Chvatal. *Linear Programming*. Freeman, 1983.

[20] L. Lasdon. *Optimization theory for large systems*. MacMillan, New York, 1970.

[21] H. Zang, C. Ou, and B. Mukherjee. *Path-protection routing and wavelength assignment RWA in WDM mesh networks under duct-layer constraints*. IEEE/ACM Trans. on Netw., 11(2):248–258, Apr. 2003.

[22] J. Suurballe. *Disjoint paths in a network*. Networks, 14:125–145, 1974.

[23] J. Suurballe and R. Tarjan. *A quick method for finding shortest pairs of disjoint paths*. Networks, 14:325–336, 1984.

[24] E. Dijkstra. *A note on two problems in connexion with graphs*. Numerische Mathematik, 1(1):269 – 271, 1959.

[25] S. De Maesschalck, D. Colle, I. Lievens, M. Pickavet, P. Demeester, C. Mauz, M. Jaeger, R. Inkret, C. Mikac, and J. Derkacz. *Pan-european optical transport networks: an availability-based comparison*. Photonic Netw. Commun., 5(3):203–225, May 2003.

5

Energy-Efficient Resource Provisioning Algorithms for Optical Clouds

"When the well's dry, we know the worth of water."

–Benjamin Franklin

Buysse, J.; Georgakilas, K.; Tzanakaki, A.; De Leenheer, M.; Dhoedt, B. & Develder, C.; *Energy-Efficient Resource Provisioning Algorithms for Optical Clouds*, IEEE/OSA Journal of Optical Communication Networks, Vol. 5(3), pp. 226-239, 2013

5.1 Introduction

ICT equipment, facilities and the processes to control this equipment consume up to 4% of the world's total energy budget, implying a considerable environmental impact in terms of greenhouse gas emissions [1, 2]. This paper addresses the energy expenditure for an integrated network and IT infrastructure that can support cloud and grid architectures. The blueprint for the Grid architecture was laid out in [3]: in analogy with a power grid, users could get access to computing power on demand. Grid customers would generally create an application, submit it using

the grid middleware, and wait until the job finishes in order to collect the results. A more commercial version, the cloud infrastructure, extends this concept and applies the Infrastructure-as-a-Service (IaaS) concept. The consumer decides on a number of Virtual Machines (VMs), which are to be deployed on real physical devices, to which access is granted during a certain time. Cloud computing is seen as an energy-efficient architecture, as end users are limited to low-power devices, while processing power (and hence also a large part of energy consumption) is moved to the cloud [1]. Moreover, cloud architectures provide aggregation points for workloads that would otherwise be run on separate devices. This means that demands can be consolidated through statistical multiplexing and hosts can be better utilized. Grid and cloud architectures both require the pooling and coordinated allocation of a large set of distributed resources and we aim to optimize their utilization to reduce the overall energy consumption. As the network prerequisites for the applications we envisage are very demanding (e.g., high bandwidth and low latency), we assume an optical circuit-switched network based on Wavelength Division Multiplexing (WDM) and thus consider an optical grid/cloud context (see [4] for a recent overview on such optical grids/clouds). We jointly optimize energy consumption of network and IT resources using a scalable algorithm by exploiting the anycast principle. Anycast reflects the idea that a user is generally not interested in the location where his workload is processed "in the cloud"), as long as the requirements (which have been set in advance by so-called Service Level Agreements, SLAs [5]) are met. Hence, freedom arises as to where to execute a job or to place a VM. This paper presents a heuristic that for a given request finds (i) an IT end point to process the request (the scheduling problem) and (ii) a route from the requesting source to that IT end point in the optical network (the routing problem). Requests arrive sequentially and we are solving the online routing problem, as opposed to the offline version (e.g., [6]), which has an a priori known request vector, expressing for each source the number of requests which need to be served. Our algorithm minimizes energy consumption by either trying to share as much active resources as possible (avoiding a startup cost for each newly activated resource) or by allowing switching-off idle resources. The remainder of this paper is structured as follows. Section 5.2 starts off with an overview of related work, where we indicate the novelty of our contribution. Next, in Section 5.3, we present our power model for the grid/cloud infrastructure (including quantified power consumption figures). In Section 5.4 we detail the routing/scheduling algorithms, which are subsequently investigated by a detailed simulation case study in Section 5.5. Final conclusions and future work are discussed in Section 5.6.

5.2 Related work

5.2.1 Optical network energy models

Optical network technology is incontestably energy-efficient. The authors of [7] present a comparison of different IP-over-WDM architectures, demonstrating that a translucent optical architecture (i.e., the optical signal is periodically regenerated by all-optical 3R regenerators) can save up to 60% of energy compared to classical technologies (e.g., where optical signal regeneration is done in the electronic IP layer). Comparable conclusions are drawn in [8–11]: optical nodes generally consume less power than electronic ones, especially optical circuit-switched architectures based on MEMS switching devices. Furthermore, it has been demonstrated that an energy-efficient network design is coincidentally a cost-efficient design since router ports play a dominant role in both energy and capital cost. In Section 5.3 we will further discuss the model for the network energy consumption based on [8].

5.2.2 IT energy models

Regarding electricity consumption of servers and data centers, [12] indicates that power usage of all servers in the U.S. accounts for a substantial fraction of total US electricity consumption, which even doubles when auxiliary infrastructure (cooling, water pumps, etc.) is included. This is the reason that our energy model takes this supporting infrastructure into consideration. The authors of [13] investigate the power properties for servers, individual racks and clusters. They also demonstrate that nameplate ratings (manufacturer's prediction of power use) have little or no value as they tend to overestimate actual peak usage which explains why we take the parameters for a server's energy consumption from real life measurements. Secondly, they investigate the influence of Dynamic Voltage Scaling (DVS): this method reduces energy consumption by slowing down the rate of CPU processing since the faster the processing rate, the higher the energy consumption. Our energy model for a server is based on this work, while we changed the model for racks and data centers using up-to-date cooling techniques. Another strategy for IT energy minimization is server consolidation. The authors of [14] have investigated this while also trying to predict which nodes will need to be powered down/on in the future. These previous ideas, i.e., server consolidation and DVS, are combined into a single formalism in [15].

5.2.3 Energy-efficient operation in optical networks

Switching off network elements to save energy has been evaluated in [16] for an offline scenario (i.e., traffic is known beforehand - as opposed to our approach).

The authors demonstrate that, for the scenarios under consideration, there is an energy saving potential of total network energy. Similar conclusions are drawn in [17], which extends [16] with an empirical study for power consumption of a router. Scaling down the logical IP topology in an IP-over-WDM network is investigated in [18]. The authors assign a higher cost for IP links having a load below a certain threshold, deviating traffic flows from these links to remove the IP links from the IP topology. Results show that a high threshold only favors architectures which make use of equipment with high idle power (e.g., as demonstrated in [17]), as for the more EE (energy-efficient) equipment longer paths (which lead to more transit traffic in core interfaces) lead to an increase in power consumption, as the power requirements are proportional with interface bandwidth. The effect of putting clusters of network nodes in a sleep state, by routing to an appropriate location (thus using anycast as described in Section 5.1), is examined in [19]. Our work differs in that we allow powering down individual network nodes, network links as well as data centers. Power-awareness combined with resiliency aspects is investigated in [20], but only considers the network resources and a unicast scenario: the authors achieve power reduction by putting network resources into a sleep state when they are used as backup resources and demonstrate the effectiveness by comparing different routing algorithms. Although in our work we do not consider protection, we are using a similar network energy model where different components of network entities can be shut down. In [21] the authors propose to groom sub-wavelength traffic into light paths, while allowing a modular network node to offer energy savings by powering on/off chassis, modules or ports depending on traffic entering the network node. They conclude that at off-peak hours, a traditional (minimizing the number of light path setups per request) and energy-aware approach have about the same energy consumption. In peak conditions however, the energy-aware approach outperforms the traditional strategy (regarding energy consumption) since more traffic requests can be routed through already active components. A comprehensive overview of ongoing research regarding energy efficiency in telecom networks, with a specific emphasis on optical technologies, is presented in [22]. For several network architectures (metro, access and core), energy minimization opportunities are investigated and related ongoing standardization efforts are overviewed. They also indicate that there might be a potential in scheduling jobs in a grid context, allowing servers to be switched off. We build on this concept, while also considering the energy consumed in the optical core network in between the IT end points and the data centers.

5.2.4 Energy-efficient operation in data centers

The work in [23] reviews methods and technologies currently deployed for energy-efficient operation of computer hardware and network infrastructure, particularly

in cloud contexts. They demonstrate that data center scheduling can influence energy consumption and that virtualization of resources can be beneficial from an energy consumption perspective. These policies only focus on one part of the cloud, either the network or the data center, but no work tries to combine both realms. The authors indicate possible improvements, such as reducing energy consumption due to communications, which is the aim of this paper. In [24] the authors investigate how to build a cluster-scale network (within the data center premises) whose power consumption is proportional to the amount of traffic it is transmitting. They demonstrate that a flattened butterfly topology (similar to a fully connected torus) operated at a data rate proportional to the offered traffic intensity of the data center, is the most energy-efficient intra data center network design. The work in [25] presents an intra data center scheduling approach (for a three-tier network) that combines energy efficiency and network awareness: it allows analyzing data received from the switches and links and takes actions based on the network feedback. The scheduling approach avoids hotspots within a data center while minimizing the number of computing servers required for a job execution (job consolidation). In our work however, we do not consider advanced intra data center scheduling of jobs, but enforce a First-Come-First-Served (FCFS) policy. Note that this work complements ours, where we do not provide detailed modeling of the intra data center network. We believe that incorporating such more advanced intra data center scheduling will not impact our qualitative discussions pertaining to the importance of jointly considering (core) network and data center energy consumption.

5.2.5 Energy-efficiency in an integrated infrastructure

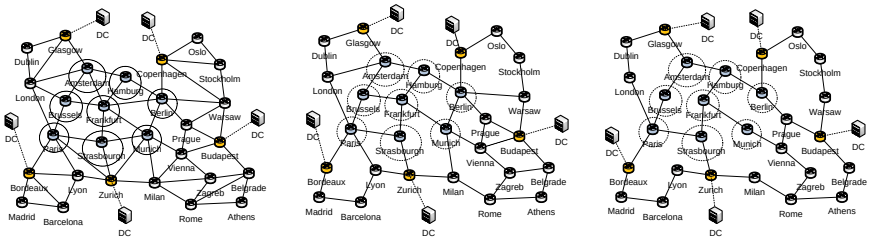
Dynamically powering on/down servers to address actual demand in a grid context has been investigated in [26]. The authors propose a power-aware scheduling scheme that reduces IT power consumption. The penalty is an increase in network utilization because longer paths are used. Our work builds on this concept by also considering the optical network, jointly optimizing the utilization of IT and network resources used to serve all demands. Chapter six of [27] proposes two ways to reduce energy consumption: (i) a novel, integrated optical network and IT infrastructure and (ii) an energy-aware service plane architecture. The first optimization consists of distributing a fraction of IT nodes from IT resource sites at the network edge into the network core so that network operators can benefit from the existing space, cooling and power of switching nodes in the core of the network. The second optimization consists of a resource orchestration formulation taking into account energy-aware parameters, such that the selection of network and IT resources is optimized to reduce the overall power consumption. Depending on the scenario, the new integrated infrastructure can improve energy efficiency up

to 45% and the EE resource orchestration up to 10%. Another attempt to define a comprehensive energy model where network and IT resources are treated in an integrated way has been examined in our earlier work [6]: it addresses the energy-efficient operation of integrated network and IT infrastructures in the context of cloud computing in an offline scenario. There, we proposed energy-efficient routing and IT allocation algorithms using MILP, by allowing switching-off several IT and networking elements and by exploiting the anycast principle. More specifically, comparing joint minimization of both network and IT energy provides energy savings of the order of 3% to 55% compared to the network energy minimization only approach, depending on the granularity of a data center to switch on/off a set of servers. On the other hand, pure network-energy minimization allows energy savings of the order of 1% to 2% of the total energy budget compared to shortest path routing (i.e., energy-unaware). Although [27] and [6] indicate that treating network and IT resources jointly allows for energy optimization, their approaches are difficult to adopt in real settings since they suffer from scalability issues and cannot produce results in a reasonable time frame. Therefore, we extend this earlier work in two ways: (i) we update the energy model to include energy-efficient cooling units (In-Row Cooling) and (ii) we tackle the problem in an online scenario to obtain results in a faster time frame.

5.2.6 Contribution of this paper

Our study extends previous works in several ways. Our first main contribution consists of the integration of the network and the IT realm: by considering optical and IT resources in the same scheduling and routing step, we lower the overall energy consumption considerably. Moreover, we provide a one-step anycast calculation and compare it with a sequential computation (two-step, first IT data center selection, then routing towards it) and show the benefits of our unified approach in terms of power consumption and service blocking. Furthermore, we allow switching off network nodes, links, servers, racks and data centers in contrast to previous works which mainly focused on either the core network or the IT infrastructure. Secondly, our unified energy model considers a cooling system, namely in-row cooling, which proves to be the most energy-efficient cooling system for data centers available today [28]. Thirdly, we treat the problem from an online perspective, as opposed to the offline scenario, resulting in an algorithm that is able to dynamically allocate resources in a short time frame. Lastly, we focus not only on energy consumption, but we also investigate the influence of EE scheduling and routing on traditional QoS parameters such as service blocking and average resource load (as opposed to e.g. [27]).

5.3 Modeling



(a) Dense topology (57 fiber links, node degree 4.03)

(b) Basic network (40 fiber links, average node degree 3.07)

(c) Sparse network (33 fiber links, average node degree 2.4)

Figure 5.1: The topologies considered in this study, containing 28 OXCs. The circled OXCs are the eight core nodes. The dotted lines between the DC's and the network nodes are the virtual links. All topologies were gathered from [29].

5.3.1 Topology modeling

We model the optical network as a bidirectional graph $G = (S, C, E)$ where S is the set of source nodes, comprising optical cross-connects (OXCs) generating requests. C is the set of core OXCs, which (as opposed to source OXCs) may be switched off completely. E is the set of optical fiber links connecting all OXCs $(S \cup C)$. Each fiber is assumed to have W wavelengths. The topologies used in our study are presented in Fig. 5.1. Furthermore we define $D \subseteq S$ as the set of destination sites, i.e., these OXCs $d \in D$ are connected to a data center. Our graph model employs auxiliary links between the data center objects and $d \in D$, which we will denote as virtual links, as they do not represent actual physical links. All fiber links incident to $d \in D$ have $2W$ wavelengths, as these are the end points of all paths and need more capacity to prevent network blocking. We assume that all data centers have the same characteristics: each data center d has n racks, each containing s servers with idle and peak power characteristics described and measured in [30].

5.3.2 Network energy modeling

We assume OXCs based on a photonic switching matrix that is realized by 3D Micro-Electrical-Mechanical-Systems (MEMS) [31]. Each OXC supports a number of input and output fibers ports, each employing a maximum number of wavelengths W. It is assumed that each OXC is equipped with wavelength converters at the output so that a light path (a wavelength path including all used wavelength links from source to destination) can be established between any source-

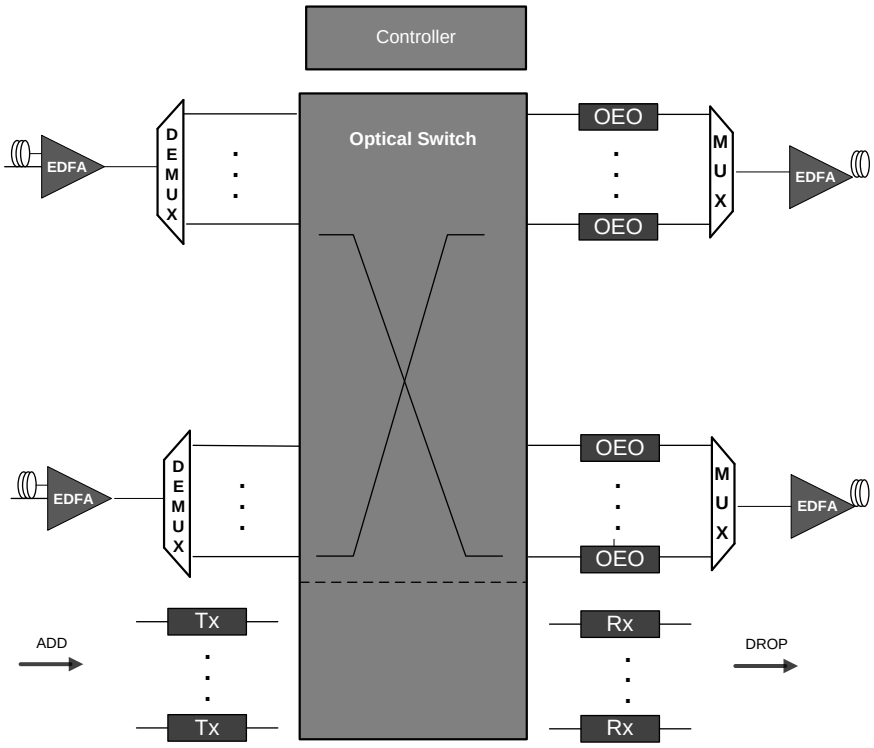


Figure 5.2: Layout of an opaque OXC. All elements except for the (de)multiplexer consume energy.

destination pair as long as there is a free port, avoiding situations of wavelength blocking. Apart from the passive elements, being the Multiplexers (MUX) and De-Multiplexers (DEMUX), Fig. 5.2 illustrates the active elements of the OXC: the switch matrix, one Erbium-Doped Fiber Amplifier (EDFA) per input/output fiber port and one transmitter (Tx) and one receiver (Rx) pair per light path. The OEO transponders support optoelectronic regeneration and full wavelength conversion. The number of through (express) ports ($ports_{through}$) is calculated as the number of input fibers times the fiber wavelength capacity W. The add/drop ports (e.g., for traffic from/to a local data center) are denoted as $ports_{a/d}$). The active incoming/outgoing fibers are represented as f_{in} and f_{out} respectively. The network power is completely determined by the power consumption of all the OXCs and the optical fiber links. The power expenditure of an OXC (P_{OXC}) depends on the constant power consumption of (i) the switch fabric (P_{sf}), (ii) the receivers and transmitters (P_{transc}), (iii) the wavelength converters (P_{conv}), (iv) the optical amplifiers (P_{ampl}) and (v) the controller power ($P_{control}$) for the OXC. Eq. 5.1 show how these figures are used in the total power consumption model of the

OXC, while Table 5.1 shows typical values for their parameters.

$$P_{OXC} = P_{control} + P_{sf} + P_{transc} + P_{conv} + P_{ampl} \tag{5.1a}$$
$$P_{transc} = ports_{a/d} \times P_{Tx/Rx} \tag{5.1b}$$
$$P_{conv} = ports_{through} \times P_{transp} \tag{5.1c}$$
$$P_{ampl} = (f_{in} + f_{out}) \times P_{edfa} \tag{5.1d}$$

Regarding the fiber links of the optical networks, the power consuming elements are the optical amplifiers installed per span. The amplifier span length ($span$) is assumed to be 80km. Hence, the power consumption P_l of a fiber link depends on its length ($|l|$) and can be calculated as shown in Eq. 5.2 (Note that -1 is used because the first span can be covered by the EDFA at the fiber output port of the OXC).

$$P_l = (\frac{|l|}{span - 1}) \times P_{edfa} \tag{5.2}$$

The total network energy consumption is then computed by Eq. 5.3. Note that we multiply the network energy with a factor called the Power Usage Effectiveness (PUE), to account for energy used for cooling and power delivery for the network resources, and typically amounts to around 2 [9]. We have chosen not to model the power delivery and cooling chain in more detail for the network. Indeed, the values for cooling and power delivery for a data center and an OXC differ in several orders of magnitudes. Hence, a more accurate power cooling model for OXCs would not change our results qualitatively (while a simple PUE approach as opposed to our current model for the data center would).

$$P_{network} = PUE \times \left(\sum_{n \in S \cup C} P_{OXC} + \sum_{l \in E} P_l \right) \tag{5.3}$$

5.3.3 IT energy modeling

5.3.3.1 Power consumption of a server

We express the capacity of a server using floating-point operations per second (FLOPS). A server's power consumption is accurately estimated by Eq. 5.4 given its current load ϕ_{server} expressed in FLOPS, its maximum processing capacity z_{server} (also expressed in FLOPS), the power in idle state P_{idle} and the power at maximum load P_{max} [13].

$$P_{server}(\phi_{server}) = P_{idle} + \frac{P_{max} - P_{idle}}{z_{server}} \times \phi_{server} \tag{5.4}$$

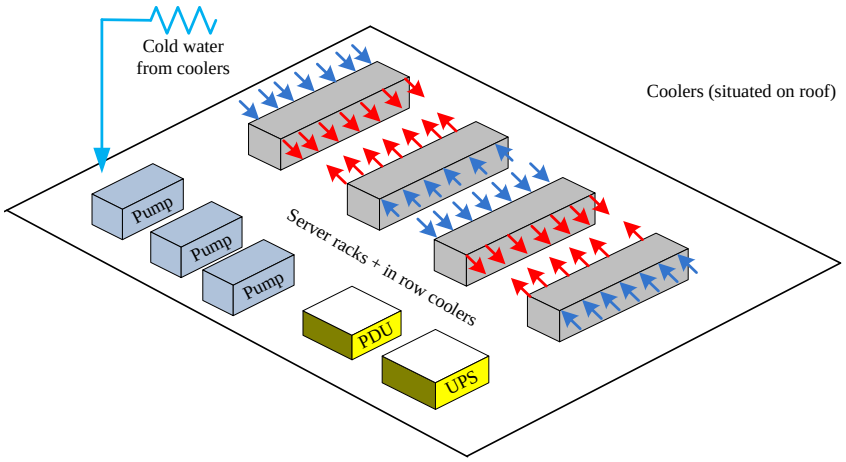


Figure 5.3: Energy consuming devices in our data center model.

5.3.3.2 Power consumption of a data center

We formalize the energy consumption of a data center based on a typical state-of-the-art deployment (see Fig. 5.3). In this model, a data center consists of rows of IT equipment which contain servers, storage devices and other supporting hardware such as coolers, water pumps (to move the cooling water) and UPS systems. All power issued to these racks first passes a uninterruptible power supply (UPS) unit which serves as a battery backup to prevent IT equipment failures in case of power interruptions. Power leaving the UPS enters a power distribution unit (PDU) that sends the power directly to the racks and servers. Note that (i) the electricity consumed by the power delivery chain (PDU+ UPS) accounts for a substantial portion of the overall power consumption of the data center (depending on the technology and load up to half of the total energy consumption) and that (ii) this power delivery chain on top of the pure IT power wastes some energy, which is mainly caused by energy loss at the UPS [32]. Another important factor in a data center regarding power consumption is air flow. The predominant architecture for delivering cooled air is raised floor air delivery from perimeter Computer Room Air Handlers (CRAH). CRAHs are placed around the room and distribute cold air through a raised floor with perforated floor tiles. This kind of architecture suffers from a couple of imperfections: (i) the distance between the cooling units and the heat source makes it difficult to remove the heat without mixing with the supply air and (ii) a considerable amount of energy is needed to drive the fans [28]. To overcome this, we consider an air-circulating solution that addresses these problems, called in-row or rack-based cooling. In this approach, the air cooling systems are integrated into a rack; it makes the air paths shorter, and significantly reduces the

power required to operate the fans [33]. We model the power consumption of such an in-row cooler, given the current capacity of all the rack's servers , the same way as a server; linearly interpolated between a P_{min}^{inrow} and P_{max}^{inrow}, as in Eq. 5.5.

$$P_{rack} = P_{min}^{inrow} + \frac{P_{max}^{inrow} - P_{min}^{inrow}}{\sum_{server \in rack} z_{server}} \times (\sum_{server \in rack} \phi_n) \tag{5.5}$$

Apart from air flow, we still need a cooling mechanism. The assumed deployment uses k dry coolers / free coolers which cool the water to about 17-18 C. Finally, the pumps that circulate the cooled water to the racks have to be accounted for. Concluding, the power consumption of our data center prototype is shown in eq. 9 while Table 5.1 shows values for these parameters, based on actual readings of the Ghent data center (which serves as our state-of-the-art example, both in technology and in dimensions) or equipment data sheets. Our model allows switching off certain parts of a data center, which gives us freedom in our request scheduling:

- When a server is not in use, we switch it off completely.
- Whenever a rack has no active servers we allow to switch off the in-row coolers
- When no racks are active we allow switching-off the coolers, pumps and UPS system (start up cost for a data center).

$$P_{DC} = P_{base} + \sum_{r \in racks} P_{rack} + \sum_{s \in servers} P_{server}(\phi_s) \tag{5.6a}$$

$$P_{base} = \begin{cases} 0 \text{ if not in use} \\ P_{UPS} + P_{pumps} + P_{cooler} \text{ otherwise} \end{cases} \tag{5.6b}$$

5.4 Provisioning algorithm

We investigate two approaches of scheduling and routing. The first algorithm is based on an integrated scheduling approach, where the destination site and the route towards that destination are found in a single pass, optimizing the network and IT infrastructure utilization simultaneously. We will refer to this approach as Full-Anycast (FA). In a second approach, we first decide where to handle the request and find the route towards that destination subsequently. This means that scheduling a request consists of two separate calculations: in a first step it optimizes the IT infrastructure, followed by the best possible routing given the IT destination. This latter approach (denoted Assisted Anycast or AA) constitutes the

Table 5.1: Parameters and power consumption figures for the network and IT resources. References are provided where possible and "AR" (actual reading) indicates that average power was measured on site at the Ghent University data center(01/2012).

Symbol	Description	Value	Ref.
S	Set of source nodes generating requests	20	
C	Set of core nodes. These do not generate requests and can be switched off completely.	8	
E	Set of bidirectional links.	56	
D	Set of OXCs which are connected to a data center.	5	
W	Amount of wavelengths per fiber link.	16	
n	Number of servers per rack	20	
π	Number of racks per data center	45	
κ	Number of free coolers	3	
P_{max}	Power consumption of a server when at 100% load.	268 Watt	[30]
P_{min}	Power consumption of a server when unused	144 Watt	[30]
P_{idle}^{inrow}	Power consumption of in-row cooler when unused.	300 Watt	AR
P_{max}^{inrow}	Power consumption of in-row cooler when all its servers are at 100% load.	500 Watt	AR
P_{pumps}	Average power consumption of the pumps for the cooling water	28500 Watt	AR
P_{cooler}	Average power consumption of the coolers.	13000 Watt	AR
P_{ups}	Average power consumption of UPS.	12500 Watt	AR
P_{transp}	O/E/O: Power consumption of a line-side WDM Transponder (10Gbit/s)	35 Watt	[34]
$P_{control}$	Power consumption of a controller	150 Watt	[30]
P_{sf}	Power consumed by the switching fabric.	30 Watt	[35]
$P_{tx/rx}$	E/O,O/E: Power consumed by either a transmitter or a receiver	5.9 Watt	[8]
P_{edfa}	Power consumption for an EDFA.	15 Watt	[34]
$span$	Amplifier span length	80 km	

state-of-the-art technique in commercial cloud infrastructures. As a last remark, both FA and AA only consider data centers still having enough capacity to fulfill the request. For both FA and AA, when a request has been scheduled to a data center, the data center enforces a First-Come-First-Served policy (FCFS): it first tries to schedule the requests to the first active server (in an active rack) it finds. Only after deciding there are no active servers that can process the request, a new rack is activated with the necessary servers.

5.4.1 Full Anycast (FA)

The FA routing algorithm uses a function $P^{FA} : (E \times \mathbb{N}) \rightarrow \mathbb{R}$ found in Eq. 5.7, for assigning link weights for link l when a request r needs to be scheduled, after which it computes the shortest path based on these weights using Dijkstra's algorithm. We assume $\phi \in \mathbb{N}$ to be the amount of requested IT capacity. Note that in Eq. 5.7b we add 1 in the sum, to also account for the EDFAs situated in the source and destination OXC of link l. $P_{DC}(l, \phi)$ only works for virtual links, i.e., the graph edges which connect an OXC with a data center. The function $P_{DC}(\phi)$ returns the additional power needed if request r were to be scheduled to data center DC. Assume we have a function $P(DC)$ which returns the current power of data center DC and $P'(DC)$ the power of the same data center after scheduling request r, then $P_{DC}(l, \phi)$ is given by $P'(DC) - P(DC)$.

$$P^{FA}(l, \phi) = \alpha.P_{link}(l) + \beta.P_{node}(l) + \gamma.P_{DC}(l, \phi) \tag{5.7a}$$

$$P_{link}(l) = \begin{cases} 0 \text{ if link } l \text{ is in use} \\ \left(\left\lceil \frac{|l|}{span} \right\rceil + 1 \right) .P_{edfa} \text{ otherwise} \end{cases} \tag{5.7b}$$

$$P_{node}(l) = \begin{cases} P_{sf} + P_{control} + P_{transp} \text{ if end of } l \text{ is inactive} \\ P_{transp} \text{ otherwise} \end{cases} \tag{5.7c}$$

$$P_{DC}(l, \phi) = \begin{cases} P_{base}(\phi) + P_{DC}(\phi) \text{ if adjacent DC of } l \text{ is inactive} \\ P_{DC}(\phi) \text{ otherwise} \end{cases} \tag{5.7d}$$

We mention that when $\alpha = \beta = \gamma = 1$, the function $P^{FA}(l, \phi)$ attributes each link the extra power it requires if that link (virtual or actual) were to be used to handle request r. By changing the values of α, β and γ, we change the relative importance of power contributions of links, OXCs or data centers, which has been shown to impact the QoS (e.g., blocking [36]). Moreover, by choosing another value than one for α, β and γ we modify the algorithm from a greedy approach to an algorithm which leaves resources in an inactive state, although the local optimum would activate them, which could be beneficial in the future. In our

performance evaluation, we will demonstrate a relation between energy consumption and QoS by changing the values for α, β and γ. In this work we will denote a parameter set as $\{\alpha, \beta, \gamma\}$.

5.4.2 Assisted Anycast (AA)

As mentioned above, the assisted anycast algorithm consists of two steps. First we select the data center to handle the request after which we find a route to that data center. We investigate four heuristics to select the data center:

- *Closest*: chooses the data center physically closest to the requesting source;
- *L-max*: chooses the data center with the highest current load (concentrating IT requests as much as possible);
- *L-Min*: chooses the data center with the lowest current load (performing IT load balancing);
- *Random*: randomly chooses a data center (as a benchmark strategy).

When assigning link weights to the graph edges, we only use the network-related terms from FA. More specifically we assign weight to the links using $P^{AA} : E \rightarrow \mathbb{R}$ found in Eq. 5.8.

$$P^{AA}(l) = \alpha.P_{link}(l) + \beta.P_{OXC}(l) \tag{5.8}$$

5.5 Performance evaluation

We will show results for the simulations performed for the dense EU topology, portrayed in Fig. 5.1, with 28 nodes, of which 8 are core nodes and the remaining 20 source nodes. Section 5.5.1 presents results assuming communication-intensive requests, while Section 5.5.2 will confirm that our conclusions hold for an IT-intensive request scenario. In Section 5.5.3 we will present results for the other topologies found in Fig. 5.1. All source sites $s \in S$ adopt a Poisson process to generate requests, with mean arrival rate λ and mean holding time μ, which accurately fits real world Grid job traces [37]. Consequently the load per source site is expressed in Erlang (λ/μ). Each request requires one bandwidth unit (i.e., one wavelength) and a fixed amount of IT capacity (which correspond to a number of servers) which needs to be provisioned at a single data center. The dense topology contains 57 bidirectional fiber links, with each link supporting $W = 16$ wavelengths, except for links to OXCs that are directly connected to nodes which house the data centers (which support $W = 32$ wavelengths). The link lengths correspond to the actual distance between adjacent vertices (cities). Each data

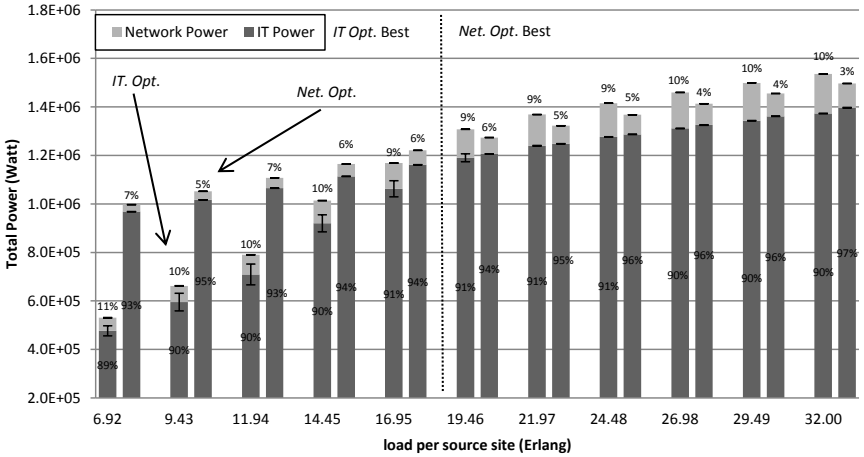


Figure 5.4: Power consumption (divided into consumption for network and IT resources) for two parameter sets: first bar IT Opt. and second bar Net. Opt. The numbers on the bars indicate the contribution of either network or IT resources to the total energy budget. Up to 16.95 Erlang, IT Opt. is best, after which Net. Opt. has lower values.

center is equipped with 20 racks, each containing 45 servers. We have performed 20 simulations (with a certain warm up period) with different seeds for every load and averaged the results; where possible the graphs show error bars, indicating the 95% confidence interval. We stopped the simulation after having served 200.000 requests. We have used a custom-built simulator [38], developed in the context of the GEYSERS [39] project.

Simulations are initially performed for a scenario where network connectivity is important (we require 3.3 servers per request) and named this the *network-intensive scenario* and later we perform the same set of simulations with identical seeds where we increase the requested IT capacity per bandwidth unit to 8.3 servers per request, which we denote as the *computing-intensive scenario*. We start with a thorough analysis of the FA algorithm, which we compare to AA in Section 5.5.1.4. The parameters α, β and γ have been ranged between 0.001 and 1 of which we show results for the most important parameter sets.

5.5.1 Network-intensive scenario (FA/Dense topology)

5.5.1.1 Pure IT vs. pure network optimization (FA/Dense topology)

In order to compare savings made by parameter sets which either emphasize network or IT power minimization, we illustrate in Fig. 5.4 the total power consumption for (i) the parameter choice with a high focus on network optimization $\{1, 1, 0.001\}$ denoted as *Net. Opt.* and for (ii) the parameter set with a large

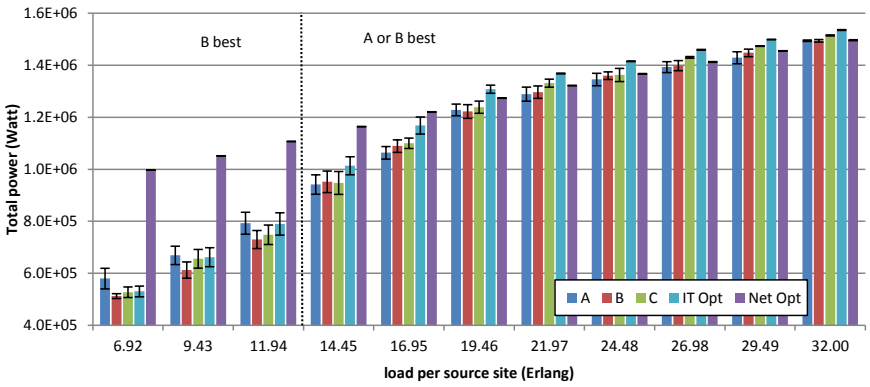


Figure 5.5: Total power consumption for parameter sets A, B ,C IT. Opt. and IT Opt. Up until 11.94 Erlang B minimizes total power consumption, after which A and B attain about the same values.

focus on IT optimization $\{0.001, 0.001, 1\}$ (*IT Opt.*) We also mention the percentage with which network and IT resources contribute to the total energy budget (depicted as the numbers on the corresponding bars), to demonstrate the balance between network and IT. Fig. 5.4 shows (i) that *IT Opt.* leads to minimal energy consumption in low load conditions while *Net. Opt.* achieves this in high load conditions and (ii) that minimizing network energy leads to an increase in IT energy and vice versa.

The large variations in total energy in low load situations (a difference up to 48%) mainly stem from switching on all data centers to optimize network power consumption for the *Net. Opt.* scenario, while fewer active data centers could serve all requests. However, starting from 19 Erlang this situation changes and *Net. Opt.* achieves a total power reduction compared to *IT Opt.* of about 3%. In these cases all data centers have to be switched on and the reduction of IT power for *IT Opt.* (on average 15.1 kW lower IT power consumption than *Net. Opt.*) is too small for the network power energy savings achieved by *Net. Opt.* (on average difference of 58.2 kW more savings in network energy than *IT. Opt.*). In what follows we will investigate how the values for α, β and γ can be chosen to lower overall power consumption even further.

5.5.1.2 Parameter set minimizing total energy consumption

Our goal is to find the parameters leading to minimal energy consumption, while keeping an acceptable level of service blocking. In this section we will investigate the influence of the parameters on the power values, while the next section discusses the effect on service blocking. The first parameter set we investigate is $\{1, 1, 1\}$, which we will denote as C. In practice, C is a greedy algorithm which

Table 5.2: Difference in total power consumption for the different parameter sets compared to absolute minimum from all simulations over all parameter sets.

Offered Load	**6.92**	**9.43**	**11.94**	**14.45**	**16.95**	**19.46**	**21.97**	**24.48**	**26.98**	**29.49**	**32**
A	11.6%	8.4%	7.9%	0%	0%	0.6%	0%	1.2%	0%	0%	0.2%
B	0%	0%	0%	1.1%	2.4%	0.1%	0.6%	2.2%	0.4%	1.3%	0.2%
C	2.8%	6.6%	2.5%	0.7%	3.3%	1.5%	3.2%	2.4%	2.6%	3%	1.6%
Net. Opt.	48.6%	41.8%	34.1%	19.1%	12.9%	4.2%	2.5%	2.8%	1.4%	1.8%	0.4%
IT Opt.	3.4%	7.5%	7.6%	7.1%	9%	6.7%	5.8%	6%	4.6%	4.7%	2.9%

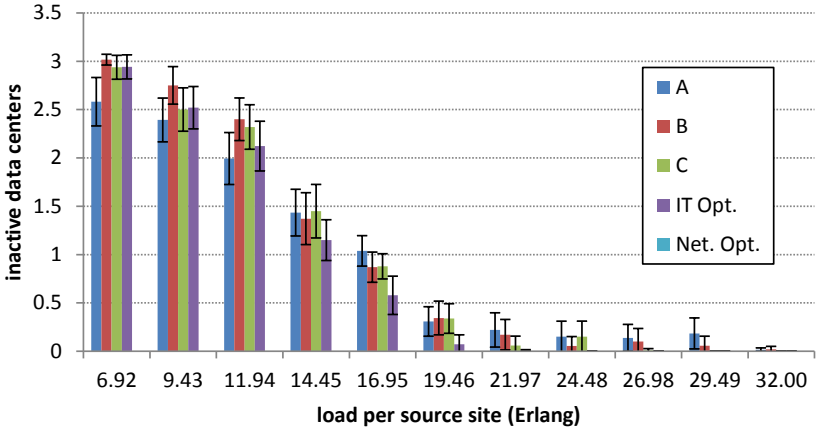


Figure 5.6: Number of data centers that are turned off. Parameter sets with a high focus for IT power minimization are clearly best in switching off complete data centers.

chooses the best routing and scheduling achievable at the moment of calculation, i.e., it chooses the local optimum. In our simulations we have performed a parameter sweep for the values for α, β and γ where we have chosen all possible combinations out of 1, $1/10$, $1/100$, $1/1000$. Results of those simulations point out two extra important parameter sets: two parameter sets with a less explicit focus on network resources than *Net. Opt.* $\{0.1, 0.1, 0.001\}$ (denoted as A) and $\{0.1, 0.01, 0.001\}$ (denoted as B). Fig. 5.5 shows the total power consumption of those parameter sets while in Table 5.2 we show the difference in power consumption for these parameter sets compared to the absolute minimum from the parameter sweep. There are three general conclusions which can be derived from Fig. 5.5 and Table 5.2: (i) in the low load range $[6.92 - 11.94]$ parameter set B achieves minimal power consumption, while in the other end either A or B is best, (ii) neither C, IT Opt. or Net. Opt. reaches this minimal power consumption and (iii) making efficient use of network resources pays off in high load conditions. In order to explain the difference in power values for each parameter set, we need to look into the ability of switching off resources, for which we refer to Fig. 5.6, Fig. 5.7 and Fig. 5.8 where we have plotted the number of inactive data centers, OXCs and fibers per parameter selection. Fig. 5.9 shows the average path length each algorithm requires.

B has minimal power consumption in the $[6.96 - 11.94]$ end, as it is more effective in switching-off data centers than A (about half a data center). The reason for this is that A sometimes reaches situations where the contribution of IT power ($\gamma.P_{DC}(l, \phi)$) is minimized to such an extent that the contributions of needed network power ($\alpha.P_{link}(l) + \beta.P_{oxc}(l)$) to reach any of the active data centers is too

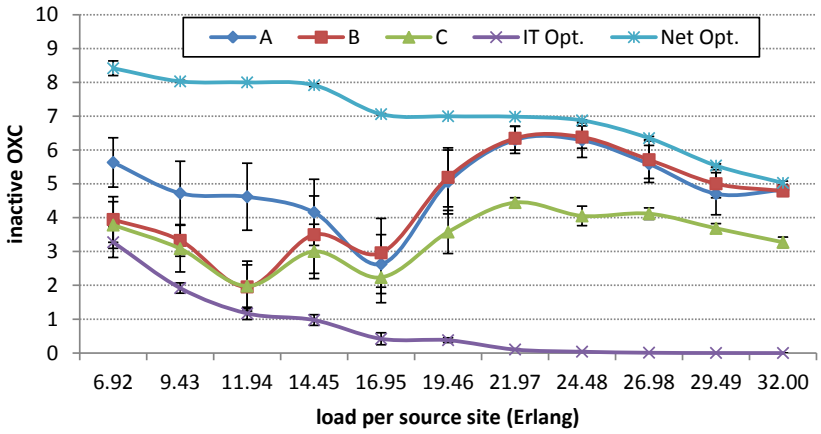


Figure 5.7: Number of OXCs which are inactive. Network focused parameters sets are switching off more OXCs. The increase around 19.46 Erlang stems from switching on all data centers (see Fig. 5.6), thus reduces the need for longer paths. Note the ability of A, B and C to turn off OXCs in higher load scenarios: as more and more data centers are turned on, the need to go through the core of the network diminishes and more OXCs can be turned off.

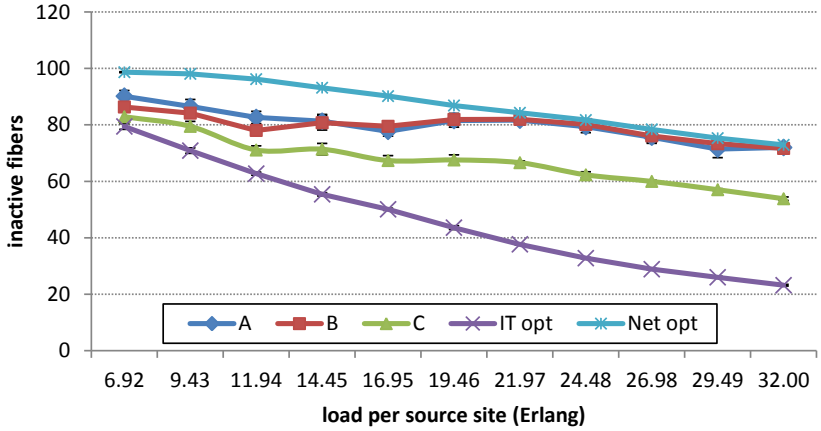


Figure 5.8: Number of fiber links which can be switched off. A, B and Net. Opt. are able to switch off significantly more fiber links.

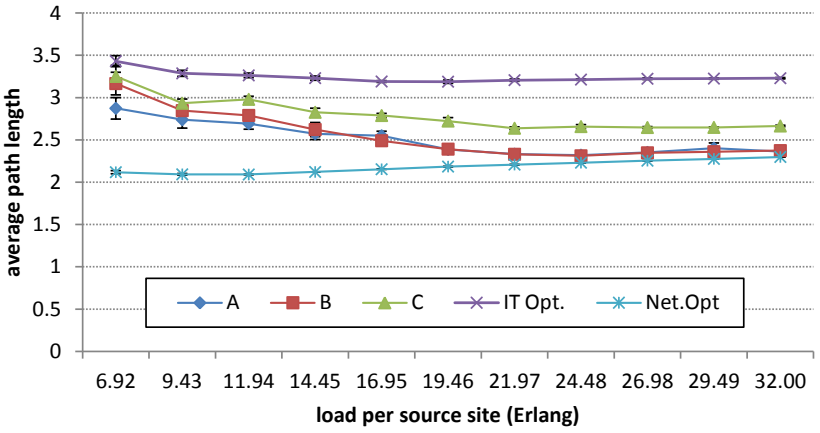


Figure 5.9: Average path length per parameter set. IT Opt. produces longer paths, to reach the IT energy minimizing datacenter.

large compared to the adjusted value for the start up cost of an inactive data center. So instead of taking a relatively long path, where additional network resources need activation, the algorithm chooses another path (using already active network resources) and boots up an extra data center. (Note that, since our algorithm does not perform a rescheduling or rerouting step at certain time intervals, this penalty stays during the complete simulation.)

Conversely, in terms of switching off network resources, A is more successful: it is able to switch off on average 2 more OXCs than B in the $[6.96 - 11.94]$ region, as it sometimes has one active data center more than B and hence shorter paths can be used (see Fig. 5.9). These network savings however do not counter the actual startup cost for the extra data center.

In the right region of the graphs ($[14.45 - 32]$) we see that A and B are able to switch off the same amount of fibers, OXCs and IT resources thus reaching about the same level of energy consumption (given that almost all data centers are active, see Fig. 5.6). As the heavy startup cost for a data center is not included anymore (only rack/server cost) in the term for IT energy, $P_{DC}(l, \phi)$, the factor $\gamma = 0.001$ minimizes the IT energy contribution to a number which is eight times smaller than the contribution of OXC power ($\beta.P_{oxc}(l)$). As paths constitute multiple hops, making β 10 times smaller (B has a $\beta = 0.01$ compared to A which has $\beta = 0.1$) does not affect the routing much and A and B reach the same routing and scheduling.

When we focus on the greedy algorithm C, Table 5.2 indicates that it never reaches the minimal total power consumption, which is also reflected in its ability for switching off resources. The intuitive reason is that C attributes the real in-

cremental power to service a new request, and does not account for the possible reuse of newly activated resources by later requests. Looking at Fig. 5.6, in the $[6.96 - 11.94]$ region, B is able to switch off a higher number of data centers. As the contribution of IT power that C accounts for is higher than that for B (or A, for that matter), longer paths are required to avoid activation of a new rack (see Fig. 5.9). As C thus requires more network resources to reach the data centers, situations occur where for a certain source node there is no (sufficient) free network capacity towards particular data centers, making it necessary to start up another data center to process the request. In the $[14.45 - 32]$ area however, almost all data centers need to be switched on in any case. Yet, for C the accounted contribution of IT power for the algorithm is still large enough (even without data center start-up costs) compared to the network resources ($P_{DC}(l, \phi)$ is about 10 times larger than $\beta.P_{oxc}(l)$ or $\alpha.P_{link}(l)$ for C). Thus, following longer paths is still cheaper with the cost metrics at hand (i.e., IT power minimization is still preferred over network power minimization). Consequently, C is unable to switch off network resources as much as A or B (see Fig. 5.7), which explains the difference in total power consumption between C and A/B.

Lastly we note that the contribution of link power (i.e., EDFAs) in the algorithm is minimal because (i) whenever a link has already been activated its contribution (as part of the algorithm) is neutralized ($P_{link}(l) = 0$) as it can be freely used and (ii) the average number of EDFAs per link is five, resulting in only an average contribution of $PUE \times (5 \times 15)$ Watt, which is small compared to the contributions of the OXCs (about 3 times when only one wavelength is routed over the OXC) and the IT resources (about 4 times for 1 rack with one server). We see that *Net. Opt.* is able to switch off significantly more fiber links than the other strategies (up to 48% compared to *IT Opt.*), as EE routing is equivalent to switching-off network resources. In low load conditions, A is able to switch off 4% more fiber links than B. As stated above, B requires this to reach better destinations to keep as much IT resources inactive as possible. Lastly, we find that *IT Opt.* is unable to switch off links as efficiently as the other strategies, as longer paths are needed to reach the best IT site.

5.5.1.3 Influence on QoS (FA/Dense Topology)

In this section we investigate the influence of parameter options on request blocking in Fig. 5.10 (due to unavailable network or IT resources), show the average network load in Fig. 5.12 and mention the data centers load. In the considered *network-intensive scenario*, there is sufficient data center capacity to meet all requests in the considered load scenarios. The only reason for requests not to be provisioned is lack of network resources, i.e., we fail to find a light path to a given server. We see that when optimizing for *IT Opt.*, we have slightly higher blocking, since paths are somewhat longer (see Fig. 5.9), thus saturating links more (see

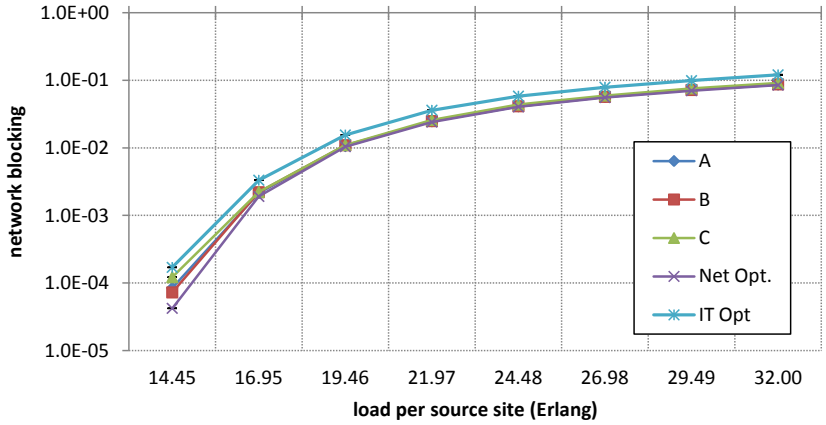


Figure 5.10: Network blocking per parameter set. Apart from IT Opt., the A, B, C and Net. Opt. have no significant differences.

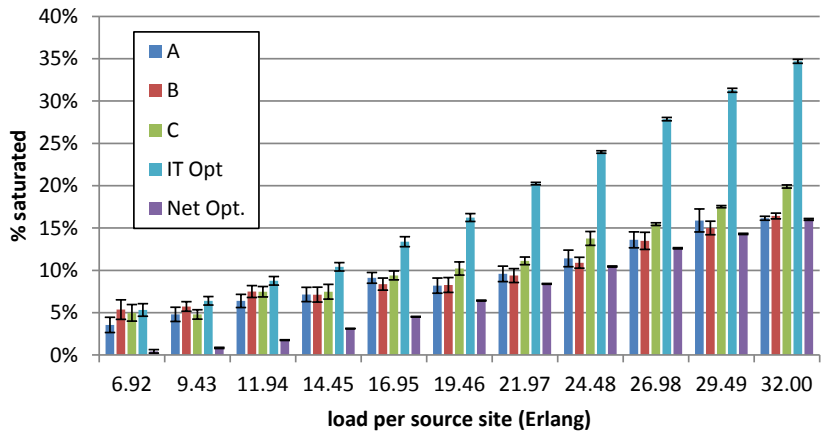


Figure 5.11: Percentage of link that has an average load higher than 85%. Parameter sets with a large focus on IT power, have a high saturation value.

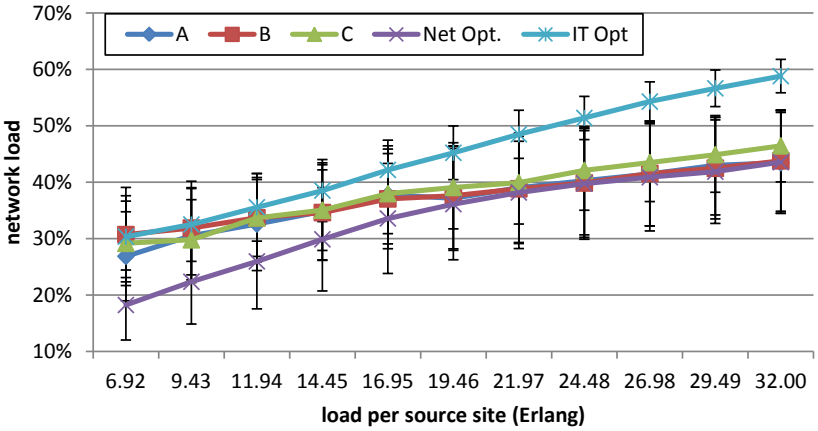


Figure 5.12: Average network load. A, B C, Net. Opt. and IT Opt.

Fig. 5.11). Differences among the other strategies are minimal.

We thus find that the strategies leading to the lowest energy consumption (A or B, see higher) are also those with lower blocking. This may sound contradictory to earlier work described in [36], showing a trade-off between energy efficiency and blocking due to network resource fragmentation resulting from long EE paths. However, this work is different in several ways. We consider a network with wavelength conversion, whereas they assume the wavelength continuity constraint. Hence, the effect of resource fragmentation on blocking in our use case is not present, as blocking only occurs when there is no capacity left anymore. Secondly, they assume a random traffic pattern, where each node of the network is a possible destination and lastly they are not switching off nodes (i.e., transponders, switching fabric, etc.) but only the optical links (EDFAs).

5.5.1.4 Difference between FA and AA (Dense topology)

In this section we study whether we can achieve the same results as the FA algorithm with a simple AA approach. In Fig. 5.13 we show the total power consumption (divided into a network and a IT portion) for the AA scheduling algorithms, with parameter settings ($\alpha = \beta = 1$). Based on results not detailed here (because of space constraints), we have concluded that the total power consumption for AA, where a destination data center site is chosen before the routing step, is hardly influenced by either α or β. The reason for this is that independently choosing the IT site, forces the algorithm to use OXCs (most dominant network resources), although another destination choice could leave the considered network resource inactive. To demonstrate the difference between AA and FA, in Fig. 5.13 we plot

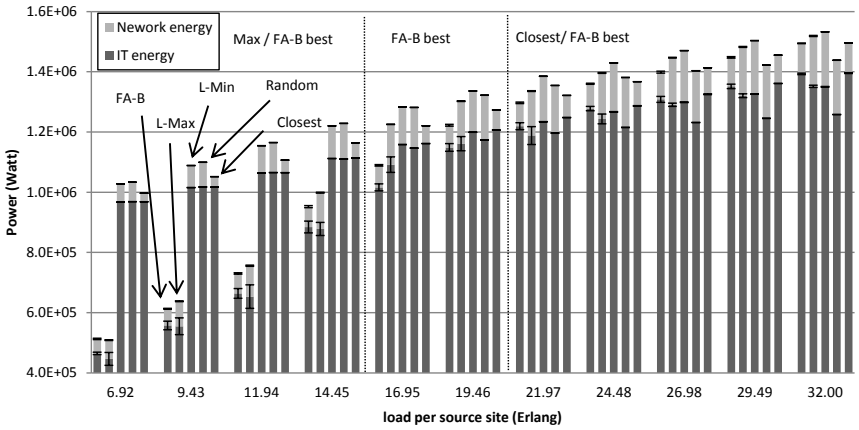


Figure 5.13: Distribution of power (network and IT energy) comparing FA (parameter set B) with AA.

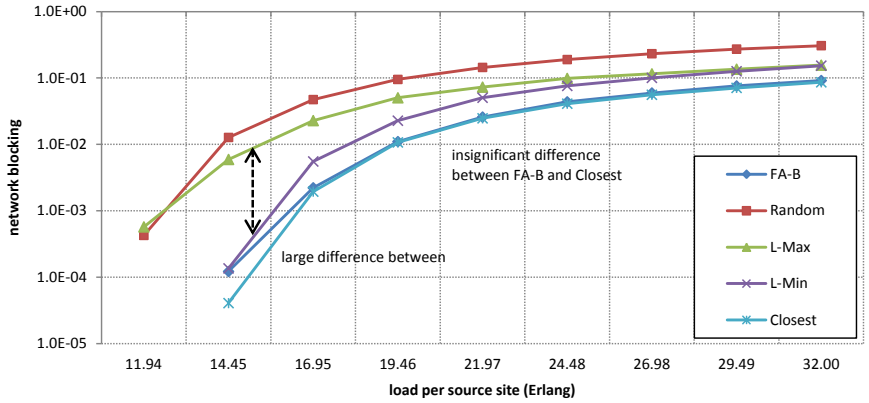


Figure 5.14: Network blocking figures comparing FA (parameter set B) with AA for different scheduling algorithms. FA-B and Closest attain about the same blocking value. L-max and Random reach inacceptable blocking figures.

the total power consumption for the AA greedy approaches ($\alpha = \beta = 1$) together with the FA values for parameter set B (FA-B). We compare the corresponding blocking probability in Fig. 5.14.

Looking at the power consumption in Fig. 5.13, we find as expected that FA-B performs best (with the notable exception of the highest loads; see our comment at the end of this subsection). Nevertheless, some AA approaches do come very close, but the exact one depends on the load region. For low loads (until 14.45 Erlang in our case study at hand), the *L-max* strategy seems the best AA approach (and only 3% above FA-B), while for higher loads *Closest* is to be preferred. The fact that *L-max* seems best at low loads is intuitively clear: in this case, it is possible to aggregate requests in a limited number of data centers (which is what *L-max* aims for) and turning off the rest. Yet, at these low loads, *L-max* leads to significantly higher blocking ratios (see Fig. 5.14) than any other AA strategy or FA-B). For higher loads (21.97 Erlang and above), intuition also expects *Closest* to be best, since there all data centers need to be powered on, and selecting the nearest data center minimizes network resource usage (and its power consumption). Network blocking for *Closest* is also similar to that of FA-B, thus making it a valid (and less complex from an implementation point of view) alternative. Only at mid loads (16.95-19.46 erlang), none of the AA approaches consumes as few power as FA-B. In conclusion, to have a single approach that attains lowest power consumption under all load conditions, none of the AA alternatives does the job, and we should resort to the FA approach.

As a final note, we mentioned that Fig. 5.13 suggests that *Random* attains the lowest total power consumption for the highest considered loads (starting from 26.98 Erlang). Yet, Fig. 5.14 shows that *Random* has a very high blocking ratio and consequently the apparent power decrease does not stem from intelligent scheduling/routing, but merely because requests are blocked and we get lower data center/network utilization.

5.5.2 Computing-intensive scenario (Dense topology)

When we increase the desired number of servers per request, we change our scenario from one where requests resemble network intensive applications (e.g., video streaming services) to applications where computation is more important. We have also simulated such a use case (for the same FA strategies with parameter settings A, B, C, *IT Opt.* and *Net Opt.*) and have reached the same qualitative conclusions as for the case described in the sections above. The power difference gap (the difference between maximum and minimum power consumption) amounts to 38%, which is 10% less than the previous use case (more power intensive IT resources need to be activated). The preference for using either parameter set A or B, depending on the load remains in the *computing-intensive scenario*: in low

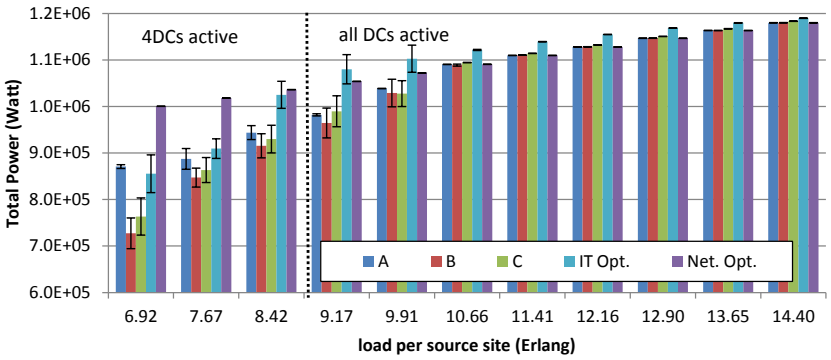


Figure 5.15: Power values for the basic network.

load conditions B is still preferred, but reaches in higher load conditions the same optimal value as parameter set A. The ability of all parameter sets to switch off network resources does not disappear, but is merely shifted to lower load conditions: from a certain point all network resources need to be activated in order to reach certain IT end points.

Although the relation for service blocking between parameters sets stays unchanged, they differ in values. In high load conditions there is not enough IT capacity left to process a request and IT blocking occurs. Although IT blocking for the *IT Opt.* parameter set is lower than for the other strategies (there is only an insignificant difference in IT blocking among A, B and C), network blocking for *IT Opt.* is prevailing, rendering the IT blocking penalty for A, B or C still small enough to outperform *IT Opt.* where total service blocking is concerned. This is also reflected in the network load, which slightly differs between *IT Opt.* and the other FA cases, leading to the difference in blocking.

5.5.3 Influence of topology

In this section we demonstrate that our conclusions for the dense topology also hold for the sparse and basic EU topology (with one major difference for the sparse topology when comparing FA and AA). Using the A, B, C , *Net. Opt.*, *IT Opt.* and the different AA algorithms, we have again performed 20 simulations and averaged the results for the basic and sparse EU topologies. The number of requested servers per request is 3.3. (We also performed these simulations for a requested 8.3 servers per request and found that the same qualitative conclusions hold.)

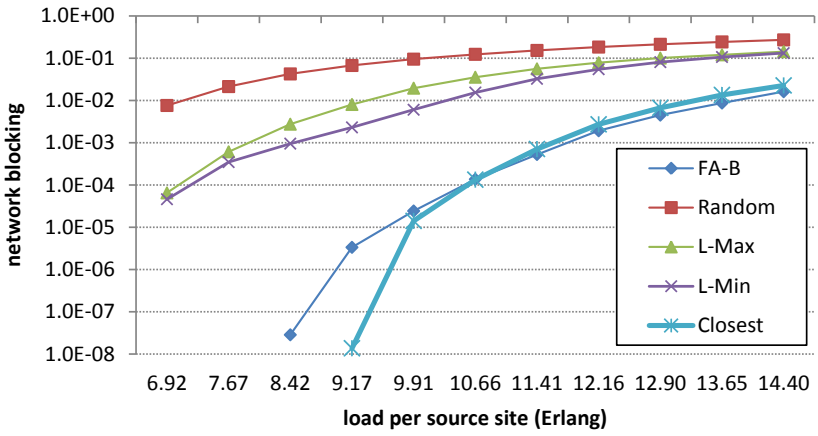


Figure 5.16: Network blocking for the basic network.

5.5.3.1 Basic topology

The difference between the basic and the dense topology is the number of fiber links (40 vs. 57). There is one major consequence with respect to energy minimization: the number of possible paths between source and destination pairs is smaller for the basic topology compared to the sparse topology. This means there are fewer opportunities for choosing a route between one of the source $s \in S$ nodes and one of the destination nodes $d \in D$. This results in (i) fewer opportunities for switching off network resources and (ii) fewer opportunities for switching off data centers as there is less network capacity. This is also reflected in Fig. 5.15 where we plot the total power consumption for the different strategies (with an adjusted load per source (λ/μ) site as we keep the number of wavelengths per link the same as for the dense topology). We conclude that all qualitative results for the dense topology also apply for the basic topology. The difference between *IT Opt.* and *Net. Opt.* is considerably lower (up to 14% compared to 48% for the sparse topology) and we see that *Net. Opt.* very quickly reaches minimal energy consumption (starting from 9.17 Erlang): all data centers need to be switched on to overcome network blocking as important links get saturated. The relative difference between A, B and C stays unchanged compared to the dense topology: in the $[6.92 - 10.66]$ region B is the best parameter choice while in the other end A, B, C, *Net. Opt.* reach almost the same optimal power consumption figures. We note that there is no significant difference for the network blocking figures for A, B, C and *Net. Opt.* and that *IT Opt.* has blocking figures which differ from the other parameter set in the orders of several magnitudes. The conclusion regarding the comparison between FA-B and AA also still applies: (i) in the low load scenario

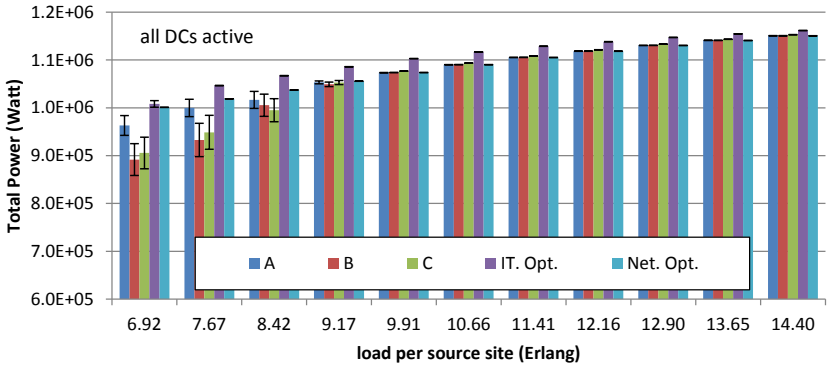


Figure 5.17: Power values for the sparse network.

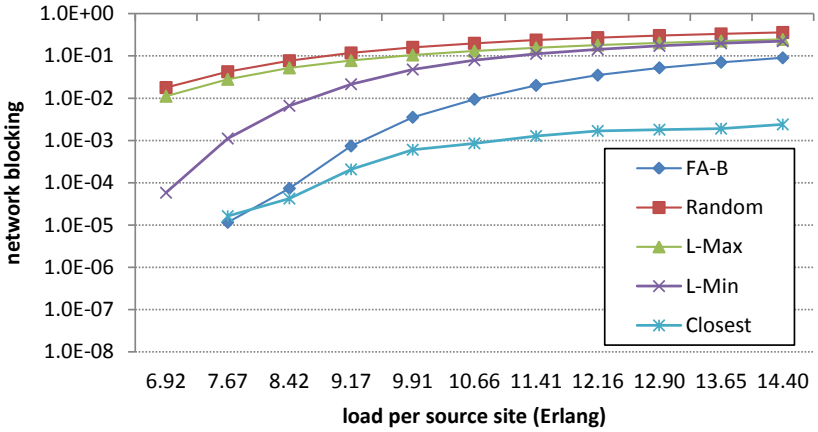


Figure 5.18: Network blocking for the sparse network.

AA L-Max approximates the FA algorithm in terms of power consumption, but with a network blocking penalty and (ii) in high load conditions FA-B has similar energy values and service blocking figures as *Closest* scheduling.

5.5.3.2 Sparse topology

The number of fiber links for the sparse topology is even less that the basic topology (33 vs. 40), thus opportunities for EE routing and scheduling are even more limited. Focusing on total power consumption (Fig. 5.17) we see that even in low load scenarios, *IT Opt.* is outperformed by the other strategies as all data centers need to be switched on to overcome network blocking. The relative difference be-

tween A, B, C and *Net. Opt.* is similar as for the basic and dense topology, with a preference for B in the low load scenarios. The relation between AA *L-max* and FA-B is also unchanged: AA *L-max* approximates FA power consumption in a low load scenario, with a service blocking penalty. In high load scenarios however, the service blocking figures for FA and *Closest* are different, although reaching the same optimal energy values. Trying to route with a power minimization objective leads to longer paths in a sparse topology. These longer paths consume precious network capacity, leading to a larger service blocking, while the power optimization seems to have no effect (compared to choosing the closest data center). The reason for the latter, is that EE routing of a single newly arriving request temporarily allows to provision it without activating new resources, but the advantage is lost quite soon when subsequent requests still require to activate new (scarce) network resources. The latter effect seems not to play in less network constrained conditions (i.e., the basic and dense topologies).

5.6 Conclusions and future directions

Energy reduction in optical networks received a considerable amount of attention in the research community. In this work, we have ported a number of ideas presented in previous works to an optical cloud context. More specifically, we have presented a unified, online and weighted routing and scheduling algorithm for a typical optical cloud infrastructure for which we have developed an energy consumption model jointly considering network and IT resources. We have performed a detailed parameter selection, which has shown that depending on the offered infrastructure load, a different selection for the weights for the resource power values is beneficial. Secondly, we have demonstrated that for topologies with a reasonable network degree, the best selection of weights on the subject of energy consumption does not lead to a service blocking penalty (apart from in highly loaded sparse networks). Lastly, we have shown that our unified full anycast algorithm, which computes the destination and route to that destination in one step, outperforms present-day assisted anycast (AA) algorithms which first consider IT resources after which routing is performed, in particular for low to medium load conditions. Possible extensions and investigations can be devised. Our scheduling algorithm only considers data center selection after which a first server selection strategy is performed over all servers and racks. Consequently, adapting the algorithm with different in-data center scheduling algorithms could lower total energy consumption even further. Another direction for future work is enforcing the wavelength continuity constraint, relieving the need for OEO conversion at OXCs (consequently lowering network energy as transponders are not necessary) and investigating different wavelength selection algorithms. Lastly, resiliency could be explored: how can we protect the integrated network and IT

infrastructure, providing resiliency for both network and IT resources, by allowing sharing inactive protection resources (links, OXCs and servers/racks).

References

[1] P. Anderson, G. Backhouse, D. Curtis, S. Redding, and D. Wallom. *Low Carbon computing a view to 2050 and beyond*. Technical report, Technol. and Standards Watch, Bristol, Nov. 2009.

[2] M. Pickavet, W. Vereecken, S. Demeyer, P. Audenaert, B. Vermeulen, C. Develder, D. Colle, B. Dhoedt, and P. Demeester. *Worldwide energy needs for ICT- The rise of power-aware networking*. In Proc. 2nd Int. Symp. Advanced Netw. and Telecommun. Syst.. ANTS, pages 1–3, Mumbai, India, 15–17 Dec. 2008.

[3] I. Foster and C. Kellelman. *The grid: blueprint for a new computing infrastructure*. Morgan Kaufmann, 1999.

[4] C. Develder, M. De Leenheer, B. Dhoedt, M. Pickavet, D. Colle, F. De Turck, and P. Demeester. *Optical networks for grid and cloud computing applications*. Proc. IEEE, 100(5):1149–1167, May 2012.

[5] P. Robinson, A.-F. Antonescu, L. M. Contreras-Murillo, J. Aznar, S. Soudan, F. Anhalt, and J. A. Garca-Espn. *Towards cross stratum SLA management with the GEYSERS architecture*. In Proc. IEEE Int. Symp. Parallel and Distributed Processing with Applications (ISPA), Legans, Madrid, Spain, 10–13 Jul. 2012.

[6] J. Buysse, K. Georgakilas, A. Tzanakaki, M. De Leenheer, B. Dhoedt, P. Demeester, and C. Develder. *Calculating the minimum bounds of energy consumption for cloud networks*. In Proc. IEEE Int. Conf. on Comp. Commun. Networks (ICCCN), pages 1–7, Maui, Hawaii, USA, 31 Jul. - 4 Aug. 2011.

[7] F. Musumeci, M. Tornatore, and A. Pattavina. *A power consumption analysis for IP-over-WDM core network architectures*. IEEE/OSA J. Optical Comm. and Netw., 4(2):108–117, Feb. 2012.

[8] A. Slaviša. *Analysis of power consumption in future high-capacity network nodes*. IEEE/OSA J. Optical Comm. and Netw., 1(3):245–258, Aug. 2009.

[9] J. Baliga, R. Ayre, K. Hinton, W. V. Sorin, and R. S. Tucker. *Energy consumption in optical IP networks*. IEEE J. Lightwave Technol., 27(3):2391–2403, Jul. 2009.

[10] G. Shen and R. S. Tucker. *Energy-minimized design for IP over WDM networks*. IEEE/OSA J. Optical Commun. and Netw., 1(1):176–186, Jun. 2009.

[11] R. Tucker, R. Parthiban, J. Baliga, K. Hinton, R. Ayre, and W. Sorin. *Evolution of WDM optical IP networks: a cost and energy perspective*. IEEE J. Lightwave Technol., 27(3):243–252, Feb 2009.

[12] G. J. Koomy. *Estimating total power consumption by servers in the U.S. and the world*. In Lawrence Berkeley National Labs, 2007.

[13] X. Fan, W. Weber, and L. Barroso. *Power provisioning for a warehouse-sized computer*. In Proc. 34th Int. Symp. of Computer architecture, SCA, pages 13–23, San Diego, CA, USA, 7–9 Jun. 2007.

[14] A.-C. Orgerie, L. Lefevre, and J.-P. Gelas. *Save watts in your Grid: Green strategies for energy-aware framework in large scale distributed Sys.* In Proc. 14th IEEE Int. Conf. Parallel and Distributed Sys.(ICPADS), pages 171–178, Washington, DC, USA, 10 Dec. 2008.

[15] Y. Chen, A. Das, W. Qin, A. Sivasubramaniam, Q. Wang, and N. Gautam. *Managing server energy and operational costs in hosting centers*. In Proc. ACM SIGMETRICS Int. Conf. on Measurement and Modeling of Comp. Sys., pages 303–314, Banff, Canada, 6–10 Jun. 2005.

[16] L. Chiaraviglio, M. Mellia, and F. Neri. *Energy-aware backbone networks: a case study*. In Proc. IEEE Int. Conf. on Commun. Workshops, pages 1–5, Dresden, Germany, Jun. 2009.

[17] J. Chabarek, J. Sommers, P. Barford, C. Estan, D. Tsiang, and S. Wright. *Power awareness in network design and routing*. In Proc. 27th Conf. on Comp. Commun. (INFOCOM), pages 457–465, Phoenix, AZ, U.S., 15–17April 2008.

[18] B. Puype, W. Vereecken, D. Colle, M. Pickavet, and P. Demeester. *Power reduction techniques in multilayer traffic engineering*. In Proc. 11th Int. Conf. on Transparent Optical Networks, pages 1–4, Sao Miguel, Azores, Portugal, 28 Jun. – 2 Jul. 2009.

[19] B. Bathula and J. Elmirghani. *Green networks: Energy efficient design for optical networks*. In Proc. Conf. on Wireless and Optical Commun. Networks. (WOCN), pages 1–5, Cairo, Egypt, Apr. 2009.

[20] A. Jirattigalachote, C. Cavdar, P. Monti, L. Wosinska, and A. Tzanakaki. *Dynamic provisioning strategies for energy efficient WDM networks with dedicated path protection*. Optical Switch. and Netw., 8(3):201–213, Mar. 2011.

[21] M. Hasan, F. Farahmand, and J. Jue. *Energy-awareness in dynamic traffic grooming*. In Proc. Conf. on Optical Fiber Commun. (OFC), pages 1–3, San Diego, CA, USA, Mar. 2010.

[22] Z. Yi, P. Chowdhury, M. Tornatore, and B. Mukherjee. *Energy efficiency in telecom optical networks*. IEEE Commun. Surveys and Tutorials, 12(4):441–458, Jul. 2010.

[23] A. Berl, E. Gelenbe, M. Di Girolamo, G. Giuliani, H. De Meer, M. Q. Dang, and K. Pentikousis. *Energy-efficient cloud computing*. The Computer J., 57(7):1045–1051, August 2010.

[24] D. Abts, M. Marty, P. Wells, P. Klausler, and H. Liu. *Energy proportional datacenter networks*. In Proc. Int. Symp. on Comp. Arch. (ISCA), pages 338–347, Saint-Malo, France, 19–23 Jun. 2010.

[25] D. Kliazovich, P. bouvry, and S. U. Khan. *DENS: Data center energy-efficient network-aware scheduling*. In Proc. IEEE/ACM Int. Conf. on Green Computing and Commun. & Int. Conf. on Cyber, Physical and Social Computing, pages 69–75, Washington, DC, USA, Dec. 2010.

[26] C. Develder, M. Pickavet, B. Dhoedt, and P. Demeester. *A power-saving strategy for Grids*. In Proc. 2nd Int. Conf. on Networks for Grid Applications (GRIDNETS), Bejing, China, Oct. 2008.

[27] C. Esther Abosi. *Towards a service oriented framework for the future optical internet*. PhD thesis, School of Computer Science and Electronic Engineering University of Essex, Apr. 2011.

[28] J. Bean and K. Dunlap. *Energy efficient cooling for data centers: a close-coupled row solution*. ASHRAE J., 1, Oct. 2008.

[29] S. De Maesschalck, D. Colle, I. Lievens, M. Pickavet, P. Demeester, C. Mauz, M. Jaeger, R. Inkret, B. Mikac, and J. Derkacz. *Pan-european optical transport networks: An availability-based comparison*. Photonic Netw. Commun., 5(3):203–225, May 2003.

[30] S. P. E. Corporation. *SPECpower*. Technical report, SPEC, http://www.spec.org/power_ssj2008/results/, 2008.

[31] A. Olkhovets, P. Phanaphat, C. Nuzman, D. Shin, C. Lichtenwalner, M. Kozhevnikov, and J. Kim. *Performance of an optical switch based on 3-D MEMS crossconnect*. IEEE Photonics Technol. Letters, 16(3):780–782, Mar. 2004.

[32] A. Katz. *Maximizing energy cost savings using high efficiency UPS*. In The electricity forum, 2006.

[33] K. Dunlap and N. Ramussen. *The advantages of row and rack-oriented cooling architectures for data centers*. In American Power Conversion White Paper, 2006.

[34] F. Idzikowski. *Power consumption of network elements in IP over WDM networks*. Technical report, Telecommunication Networks Group (TKN), TU Berlin, 2009.

[35] M. Murakami and O. Kazuhiro. *Power consumption analysis of optical cross-connect equipment for future large capacity optical networks*. In Proc. 11th Int. Conf. on Transparent Optical Netw. (ICTON)., pages 1–4, Sao Miguel, Azores, Portugal, 28 Jun. – 2 Jul.. 2009.

[36] P. Wiatr, P. Monti, and L. Wosinska. *Green lightpath provisioning in transparent WDM networks: pros and cons*. In Proc. 4th Int. Symp. on Advanced Netw. and Telecommun. Sys. (ANTS), pages 10–12, Mumbai, India, Dec. 2010.

[37] K. Christodoulopoulos, E. Varvarigos, C. Develder, M. De Leenheer, and B. Dhoedt. *Job demand models for optical grid research*. In Proc. 11th Int. IFIP TC Conf. on Optical Network design and modeling (ONDM 2007), pages 127–136, Athens, Greece, 29–31 May 2007.

[38] M. De Leenheer, J. Buysse, K. Mets, B. Dhoedt, and C. Develder. *Design and implementation of a simulation environment for network virtualization*. In Proc. 16th IEEE Int. Workshop Comp. Aided Modeling, Analysis and Design of Commun. Links and Netw. (CAMAD), pages 87–91, Kyoto, Japan, Jun. 2011.

[39] E. Escalona, S. Peng, R. Nejabati, D. Simeonidou, J. Garcia-Espin, J. Ferrer, S. Figuerola, G. Landi, N. Ciulli, J. Jimenez, B. Belter, Y. Demechenko, C. de Laat, X. Chen, A. Yukan, S. Soudan, P. Vicat-Blanc, J. Buysse, M. De Leenheer, C. Develder, A. Tzanakaki, P. Robinson, M. Brogle, and T. Bohnert. *GEYSERS: generalised architecture for dynamic infrastructure services*. In GEYSERS: A novel architecture for virtualization and co-provisioning of dynamic optical networks and IT services, pages 1–8, Warsaw, Poland, Jun. 2011. Available from: http://www.geysers.eu/.

6

NCP+: an integrated network and IT control plane for cloud computing

Buysse, J.; De Leenheer, M.; Develder, C.; Miguel Contreras, L. & Landi, G.; *NCP+: an integrated network and IT control plane for cloud computing*, submitted to Journal of Optical Switching and Networks, 2012

6.1 Introduction

The adoption of cloud computing, as a manifestation of the "utility computing" idea suggested back in 1961 by J. McCarthy, can be seen as a next step in an evolution to gradually push functionality further into the network, enabled by the evolution of e.g., optical networking (which meets high bandwidth and low latency requirements of applications varying from consumer-oriented, over more stringent business-driven to scientific cases) [1]. This shift to network-based solutions not only has benefits for users (no local configurations, automatic backups, etc.) but is also interesting for the network and service provider: updates and improvements are simpler because instead of pushing software updates to the users, the service provider only needs to update the software copy in the data center. Moreover, these services can run at a low cost as IT resources can be shared among many users.

The introduction of cloud computing, the observation that network interactions evolved from point-to-point to interworking of many distributed components and

the fact that some services may involve multiple geographically dispersed data centers (e.g., replicas to improve throughput and reduce latency), imply the need for the service provider to offer joint provisioning of IT resources at multiple data center sites as well as their interconnection. Isolated and statically interconnected data centers are evolving towards warehouse scale computing data centers [2] where network connectivity cannot rely on traditional transport technologies [3]. Optical networks with Wavelength Division Multiplexing (WDM) are the ideal candidate for the required low-latency and high-bandwidth network connectivity.

Traditionally, a network provider aiming to provide cloud services is required to make substantial investments in the integration of the variety of platforms operating over the heterogeneous resources within his management system. Thus there is a need for an automated and combined control mechanism for IT and network resources to ensure service continuity, efficient use of resources, service performance guarantees, scalability and manageability. These goals can be achieved either by reusing and combining existing separate IT and network management systems, or by developing new joint platforms. However, the former solution still implies substantial human intervention, and the efficiency of the whole system is bounded by the limits of the separate components [4]. Instead, deploying an integrated control plane would enable both scalability and efficient operation over the network and IT infrastructure, which is what this paper presents.

Note that in a cloud context, it is common to have multiple data center sites offering the same functionality (cf. aforementioned replication), which implies the flexibility to choose the most suitable resource(s). This model amounts to anycast routing, solving the problem of selecting a route to one target destination chosen from a set of nodes, as opposed to unicast where the source-destination pair is known in advance. Anycast has been shown to be beneficial for the overall network performance, either in terms of network survivability [5] [6], impairment avoidance [7], energy minimization [8, 9] or blocking probability reduction [10]. Consequently, the control mechanism of an integrated network and IT infrastructure to select both the data center (IT end point without initially specifying its location) and the network resources (the path to the chosen IT end point) becomes critical for guaranteeing an efficient operation of the entire infrastructure. In this paper we refer to such a service as a Network and IT Provisioning Service (NIPS), where IT capacity is dynamically requested in combination with the network services among the selected sites, with a network capacity tailored to the real-time application requirements.

This paper introduces a set of extensions to a Generalized Multi-Protocol Label Switching (GMPLS) [11] and Hierarchical Path Computation Element (H-PCE)-based [12] network control plane, referred to as NCP+ , to enable anycast path computation for NIPS requests in a multi-domain optical scenario, comprising also the IT resources (i.e., servers). The contribution of this paper is threefold:

(i) we propose the enhancement of GMPLS/H-PCE modules to disseminate and process IT resource information in the NCP+, both in terms of routing and signalling functionalities, (ii) discuss the main extensions to the existing Path Computation Element Protocol (PCEP) to disseminate the IT resource information and (iii) propose joint network and IT path computation and topology representation algorithms used by the PCEs and evaluate them in simulation case studies.

The remainder of this paper is structured as follows. In Section 6.2 we discuss related work, while in Section 6.3 we introduce the NCP+ architecture, propose new modules and PCEP protocol extensions in order to disseminate IT resource information. In Section 6.4 we lay out the options for topology representation, routing and resource allocation algorithms. In Section 6.5 we present our simulation analysis demonstrating the effectiveness of the proposed NCP+ algorithms in terms of service blocking. We summarize our final conclusions in Section 6.6.

6.2 Related work

6.2.1 Converged network and IT control architectures

Prior attempts to an architecture managing and controlling network and IT resources simultaneously, mainly stem from the grid computing world. In grid computing users create applications (jobs) which are scheduled to some server. Many of these jobs also require network bandwidth (large transfers of data) and consequently a network path needs to be reserved between several source-destination pairs. Cloud computing builds on this concept (similar coordination of resources is required), but manifests itself in more commercially oriented scenarios [1]. A key characteristic of cloud computing is its scalability: cloud providers virtualize their resources. This enables them to operate the infrastructure cost-effectively (avoid overprovisioning), to migrate virtual machines to other servers (making relocation possible [5]) and to share resources in a safe way. Working from the bottom up, there are three models for cloud computing: (1) Infrastructure-as-a-Service (IaaS), (2) Platform-as-a-Service (PaaS) and (3) Software-as-a-Service (SaaS)

With IaaS, companies rent the network and IT resources, with pre-loaded operating systems and barely anything else (these resources are sometimes provided as virtualized resources). IaaS users then load their own applications and platforms. In SaaS on the other hand, Cloud providers really offer a complete application that is directly usable by the consumer In between we find PaaS, where consumers rent infrastructure with a development platform which enables them to create SaaS services. Our integrated network control plane has been developped for the IaaS paradigm: once the resources have been reserved, how can we efficiently control and provision services on them?

A control plane able to control an integrated network and IT infrastructure,

requires three components: (1) a signalling component to set up the service, (2) a provisioning strategy in order to choose IT end points and routes to these resources and (3) the control/management plane to glue evertyhing together and provide the service. In what follows we will investigate prior attemps for such control systems and compare them to our NCP+ proposal, using the requirements above.

One attempt for an integrated control plane spanning both the network and the IT resources in the context of grid computing has been created by the Phosphorus project [13]. Phosphorus created an enhanced version of the ASON/GMPLS control plane to both monitor and co-allocate network and grid resources (denoted as Grid GMPLS or G^{2}MPLS). Basically, it represents the grid resources as network nodes with special capabilities and distributes this extra grid information the same way as the network resource information using OSPF-TE. G^{2}MPLS adopts a fully distributed path computation architecture: computations are performed by the ingress GMPLS controller (where the request originates), which means that all GMPLS controllers need to have the full view of the infrastructure, implying the need to disseminate OSPF-TE messages across all controllers. This contrasts with the hierarchical PCE architecture we adopt for the advertisement of the IT resources: since the path computation is centralized on dedicated PCE servers, resource updates are limited to these entities and not flooded among the GMPLS controllers (see Section 6.3). Moreover, our NCP+ provides a cloud-oriented web service interface, with a centralized access point for all the NCP+ provisioning services (service request, service monitoring and IT advertisement). This interface follows the paradigms of the REpresentational State Transfer (REST) model, commonly adopted in the current cloud environments and, in contrast with the Phosphorus approach, hides all the complexity of the internal network protocols, like OSPF-TE. This aspect is fundamental to allow an easy integration of the NCP+ into existing management systems for cloud resources and infrastructures. Finally, the semantics used to represent the IT resources within the NCP+ derives from the more recent Open Cloud Computing Interface (OCCI) [14] standards as opposed to the grid-oriented Grid Laboratory Uniform Environment (GLUE) standards [15] used in Phosphorus.

The EnLIGHTened (ENL) Computing Project had the same objective as the Phosphorus project: create an environment able to dynamically request any kind of resource (e.g., computers, storage, instruments and high-bandwidth network paths). It is based on the Highly-Available Resource Co-allocator (HARC) [16], which is an open-source system that allows clients to reserve multiple distributed resources in a single step as if they were one resource (known as atomicity). The general architecture consists of *Clients* which generate resource co-allocation requests, *Acceptors* which make the reservations and lastly *Resource managers* which talk to local schedulers for each resource. There are two important Resource Managers: (i) Compute Resource Manager which communicates with some batch

scheduler (e.g, Moab, Torque or Maui) and (ii) Network Resource Manager [17] which sends commands (with the fully specified route in an Explicit Route Object) to the GMPLS controller at the path's ingress point.

The G-Lambda project provides a standard web service interface between applications and the grid resource managers and network managers provided by existing network operators. In essence, the proposed architecture works with a central Grid Resource Scheduler (GRS) which accepts requests specifying the required number of CPUs and bandwidth. The GRS then sends its commands to two entities: (i) Computing Resource Managers which reserves the computing resources and (ii) Network Resource Management System which provisions the network paths using GMPLS.

For both the G-Lambda and ENL project, multi-domain connections are computed as follows: a local network resource broker communicates with the other resource brokers of the other domains to ask for network path quotations [18]. There are no state updates between the resource brokers (only for the inter-domain links) and consequently, the inter-domain path and end-points at the domain boundaries must be known beforehand. Hence, network optimization is limited to intra-domain path computations, as the inter-domain paths are fixed. Our NCP+ uses a hierarchical PCE architecture, where an abstracted view of the infrastructure is known by a central entity. Based on this abstracted information, both the network end-points and the paths towards them are computed, optimizing the complete infrastructure.

6.2.2 Path computation methods

Path computation in the NCP+ is performed by dedicated PCEs [12]. A PCE holds topology information and can be queried by a Path Computation Client (PCC) to determine end-to-end paths. The PCE typically stores its information in a Traffic Engineering Database (TED) and uses this information to perform constrained path computation. PCE-based path computation has been extensively studied [19], especially to facilitate inter-domain service provisioning. The PCE proposals for inter-domain path computation for unicast requests (e.g., where both source and destination are explicitly and univocally specified) can be classified into three categories, referred to as Per-Domain PCE (PD-PCE), peer-to-peer PCE (P2P-PCE) and hierarchical PCE (H-PCE). In what follows we will explain them and explain which options best fits the anycast paradigm.

6.2.2.1 Per-Domain PCE path computation

In a Per-Domain approach [20], the route to a destination in another domain is fixed (pre-computed or based on operator policies). For inter-domain path computations, each path segment within a domain is computed during the signaling process by

each entry node of the domain up to the next-hop exit node of that same domain. When an entry border node fails to find a route, boundary re-routing crankback signalling [21] can be used: a cranckback message is send to the entry border node of the domain and a new exit border node is chosen. This mechanism has several flaws: (i) the PD solution starts from an already known domain sequence which does not allow to optimize the complete infrastructure, (ii) this method does not guarantee an optimal constrained path and (iii) the method may require several crankback signaling messages, thus increasing signaling traffic and delaying the LSP setup. Consequently, PD path computation is not a suitable method for anycast path computation in a multi-domain, optical network scenario.

6.2.2.2 Backward Recursive Path Computation

The P2P-PCE architectures use the Backward Recursive Path Computation (BRPC) algorithm [22] to compute paths in a multi-domain scenario. The PCEs are provided with a pre-configured domain chain (based on agreements between infrastructure operators) and then create a Virtual Shortest Path Tree (VSPT) from the destination to the source. The VSPT is initialized from the ending node in the destination domain and is extended to all border nodes which are connected to the upstream domain in the provided domain chain. The process is repeated recursively by each domain up to the source domain, which can compute the optimal end-to-end path. A comparison between PCE based path computation techniques and signaling based path computation schemes such as RSVP and RSVP-with-crankback is presented in [23]. The use of this technique is limited for dynamic anycast path computations: multiple domain paths need to be considered (there are multiple IT end points spread over multiple domains) which would result in the concurrent computation of multiple VSPTs, where the path segment computation is not coordinated from a central entity and cannot be optimized.

6.2.2.3 Hierarchical PCE

Another approach to compute multi-domain paths in the network, which is used in our proposal for the NCP+, is to have a Hierarchical PCE (H-PCE) architecture which is similar to the routing area hierarchy as proposed in the Private Network-to-Network Interface (P-NNI) [24]. In H-PCE, a parent PCE is in charge of coordinating the end-to-end path computation, through multiple node-to-node intra-domain requests to its child PCEs located along the candidate inter-domain path. In H-PCE, (i) the domain path is not required to be pre-configured since it can be dynamically computed by the parent PCE itself, (ii) no critical information is required for inter-domain LSP calculation and (iii) no sharing of intra-domain information (topology, policies, etc.) with other domains is necessary. Consequently, H-PCE suits the anycast routing model perfectly. We note that there are several

ongoing projects which employ H-PCE as a key element for the path calculation in both management and control plane architectures: ONE [25], MAINS [26] and STRONGEST [27] are some of these H-PCE related projects which are pushing PCE based architectures, extending PCEP protocol and pushing the standardization in several forums. However, the use of H-PCE for anycast path computation in multi-domain infrastructures comprising both network domains and IT end points has not yet been investigated.

6.2.3 Topology aggregation Techniques

In H-PCE provisioning, the child topologies are abstracted into a single aggregated topology. This mechanism is mainly used to overcome scalability and confidentiality issues, which are inherent in the multi-domain scenario. Previous works that addressed the aggregation problem provided the following taxonomy [28]: (i) *Single Node* aggregation where a domain is represented by a single node, (ii) *Star* aggregation where the domain is characterized by a star and (iii) *Full Mesh* (*FM*) aggregation where the domain is represented by a full mesh topology of its border nodes.

The work described in [29] compares the single node aggregation with *FM* and investigates two wavelength selection schemes, showing a notable reduction in light path blocking using *FM*. The authors of [30] have compared the three topology abstraction mechanisms, in terms of blocking probability, network load and inter-domain connection cost. Apart from these aggregation schemes, they also propose a hybrid form where, depending on the ratio of border nodes to the total nodes of a domain, *FM* or *Star* aggregation is used. Results confirm that independent of traffic intensity, single node aggregation leads to an intolerable service blocking. The *Star* mechanism performs better then single node, but cannot achieve the same efficiency as *FM*. In high load scenario's, all aggregation schemes achieve the same amount of blocking, while *FM* is still able to achieve a higher channel occupation. Based on these works we have opted to use *FM* and *Star* as possible candidates for the aggregation techniques for the integrated network and IT infrastructure of the NCP+. We finally refer to a comprehensive survey [19] of inter-domain peering and provisioning solutions.

6.3 NCP+ architectural model

6.3.1 General overview

To date, GMPLS [11] is one of the de facto control planes, widely applied in today's access to backbone networks. The GMPLS control plane is divided into two components: (i) the signaling protocol (i.e., RSVP-TE [31]) used to reserve network resources along a path and establish connections in the transport network

and (ii) the routing protocol (i.e., OSPF-TE [32]) used to announce the resource capabilities and availabilities in the network. The general NCP+ architecture is depicted in Fig. 6.1. We organize the IT resources in multiple "IT domains", called IT Sites (IT-S). Each IT-S includes different types of IT resources that are locally controlled through an IT Manager (IT-M) (e.g., OpenNebula[1]) and we assume that all the IT resources belonging to a single IT-S are connected to a single network domain. The IT-M is the entity in charge of the management of the IT resources and is able to interact with a new component, called the NIPS Client. It is responsible for triggering the procedures for provisioning the network resources associated to the cloud service. In particular, the NIPS Client translates the description of the requested service to a set of requirements compliant with the Service Level Agreements (SLAs) established between the operator and its customer. These requirements are propagated to the NIPS Server, which acts as a centralized service access point for each network domain on the NCP+ side and triggers the necessary actions to establish the connectivity service (path computation, signaling, etc.). Note however, that the IT-M and the GMPLS control plane remain responsible for the final "configuration" of the resources in their own scope: the IT-M generates the commands to configure the IT resources, while the NCP+ manages the commands on the network side. This principle is based on the fundamental requirement to limit the impact required by the integration of the NCP+ functionalities on existing IT-Ms, which will facilitate the adoption of the solution in already deployed IT environments.

6.3.2 NIPS Client and NIPS Server

The interaction between the IT-S and NCP+ takes place in the NIPS User-To-Network-Interface (NIPS UNI) [33]) as shown in Fig. 6.2, which is a REpresentational State Transfer (REST) service-to-network interface based on HTTP between the NIPS client and the NIPS Server. The NIPS UNI enables the NIPS Client to request enhanced transport network connectivity services, receive notifications about the status of the established services and advertise the capability and availability of the local IT resources. This approach limits the impact on existing cloud middleware, since it only requires the introduction of a specific client to request transport network connections. The complexity of the network side protocols, for signaling and routing, is completely transparent for the IT-M, since it is entirely managed within the NCP+. On the NIPS server side, the REST messages are translated into the related network protocol messages (e.g., for signalling). The NIPS Server implements a UNI-Client (UNI-C) to interact with the transport network GMPLS controllers through their UNI-Network (UNI-N) component. Fig. 6.2 shows that the NIPS server, implementing the UNI-C, serves as a sort of proxy between the

[1] http://opennebula.org/

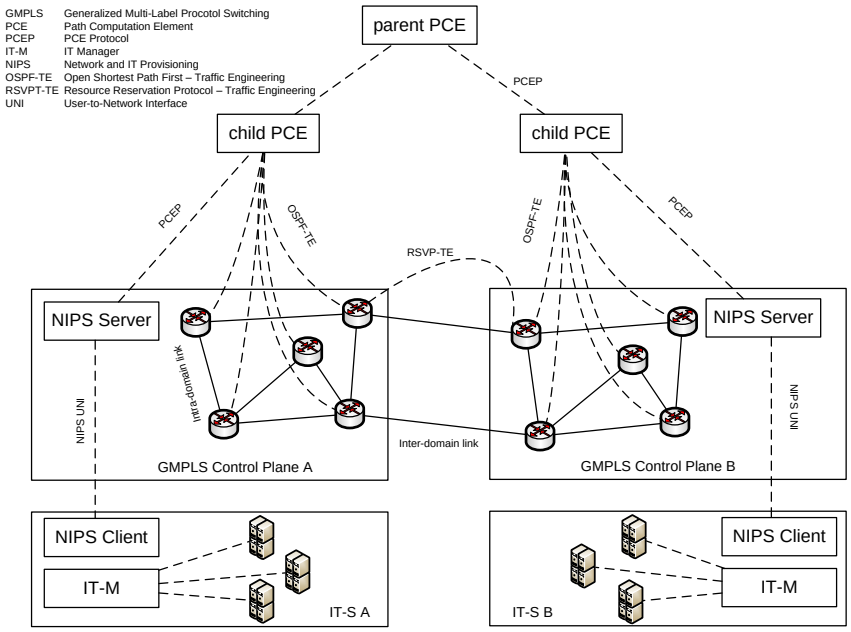


Figure 6.1: Overview of the modules of the proposed NCP+. The dotted lines represent modules interacting with each other, using the stated protocol or interface.

NIPS client and the UNI-N of the GMPLS controller. Consequently, all the service requests issued by the NIPS Client over the NIPS UNI are translated into RSVP-TE messages and propagated to the UNI-N of the corresponding GMPLS controller (i.e., the ingress node where the IT-S is attached to). For the advertisement of IT resources towards the NCP+, the NIPS server implements a Path Computation Client (PCC) to push the received information into the associated routing controller (i.e., the child PCE) on the NCP+.

6.3.3 IT resource advertisement

In order to enable inter-domain IT-aware path computation, an advertisement mechanism has to be defined to propagate the IT resource availabilities within each network domain (i.e., at the child PCE level) and at the parent PCE. In the specific case of anycast service provisioning, the choice of the IT end points of the connectivity service is made by the parent PCE. In the NCP+ architecture shown in Fig. 6.1 we propose the PCEP Notify message to update IT resource information: the child PCE collects IT advertisements in the form of PCEP Notify messages sent by the PCC of the NIPS Server. The same mechanism is in place between the child PCE and the parent PCE. With that information in place, the parent PCE can

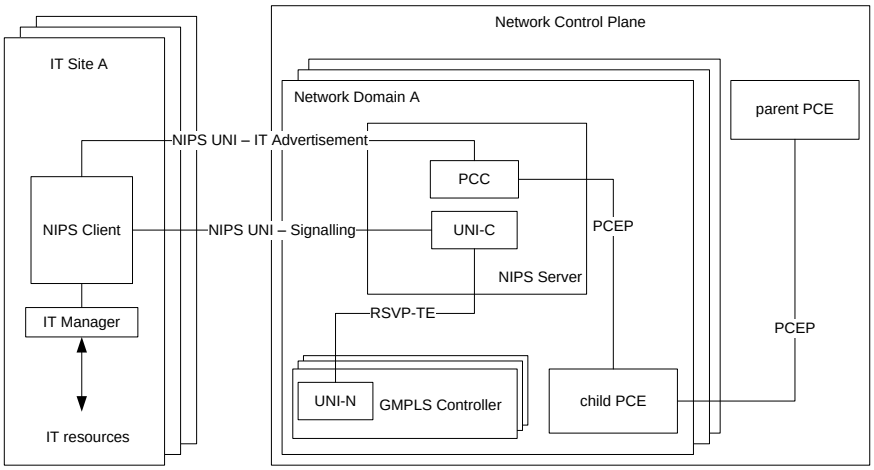


Figure 6.2: Interfaces and signaling between different modules in the NCP+.

take into account not only network Traffic Engineering (TE) parameters, but also additional attributes describing capabilities and availabilities of the IT resources. On the other hand, the network parameters are sent using the conventional OSPF-TE protocol from the GMPLS controllers to the child PCE, and from this child PCE to the parent PCE.

6.3.3.1 PCEP notify protocol extension

In the NCP+, advertisements about IT resources capabilities and availabilities at the IT-S are notified from the NIPS server to the PCE(s) through PCEP Notify (PCNtf) messages properly extended with a set of new Type-Length-Values (TLV). When the NIPS Server receives a new IT advertisement over the NIPS UNI, the PCC implemented within the NIPS Server generates a new PCNtf message that is sent to the child PCE responsible for path computation within the local domain, which in turn forwards the received PCNtf to the parent PCE.

The extended PCNtf messages for IT advertisement are compliant with the generic format described in [34]. The following new value for for the Notification Type (NT) is defined: `IT resource information (0x04)`. For the Notification Values (NV) we have defined three new values, listed in Table 6.1. The description of the IT resources is included in the optional TLV, using three new top-level TLVs, shown in Table 6.2. The Storage and Server TLVs are structured in further sub-TLVs (details see [35]). Each of them describes specific characteristics of the resource and can be structured in sub-TLVs themselves. This approach provides a flexible mechanism to notify a partial description of the resources, fundamental in case of resources updates involving a limited set of parameters or

Table 6.1: New notification values used in the extensions of the PCEP protocol.

Name	Descritpion
QUERY	sent from a parent PCE server to a child PCE and used to query all the IT resource information stored at the child PCE.
UPDATE	sent between modules which need to update information for a new or modified IT resource.
DELETE	sent between modules which need to delete an existing IT resource.

Table 6.2: New TLV values for the PCEP extention.

Name	Description
IT TLV	provides generic information about the overall advertisement, including the identifier of the originating IT site, the type of advertisement (i.e., resource add, remove, or update), and its trigger (e.g. synchronization, resource failure, resource modification from administration, etc.).
Storage TLV	describes the parameters associated with storage resources (e.g., Storage Size).
Server TLV	describes the parameters associated with a server (e.g., Operation System, Memory).

when the IT operator restricts the range of information to be disclosed to the network operator.

6.4 Path computation

The path computation function is a key feature to automate efficient provisioning of both computing resources and network connectivity tailored to the service needs. As already indicated, the NCP+ adopts a hierarchical architecture where a parent PCE is in charge of coordinating the end-to-end path computation through multiple intra-domain requests to its child PCEs. In our H-PCE model there is one centralized point (the parent PCE) maintaining a global view (without internal details) of all the domains, including the attached IT-S. These domains in turn all have a child PCE server to compute intra-domain paths. As can been seen in Fig. 6.3, the end-to-end path computation is triggered by a NIPS service request sent by the NIPS client. Upon receiving a NIPS request, the NIPS server notifies its collocated PCC, which sends a path computation request (PCReq) to the child PCE that is located in the same domain. This initially computes paths from the requesting node to all of the domain edge nodes after which it sends a PCReq message to its parent PCE. The parent PCE computes a candidate inter-domain

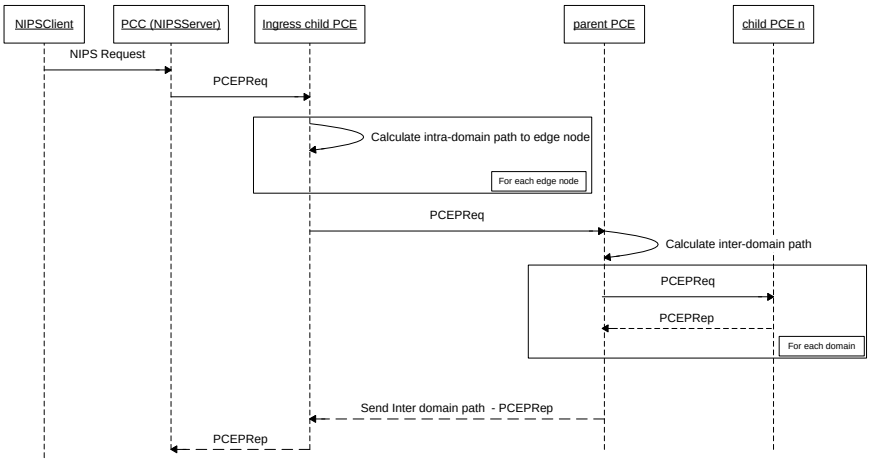


Figure 6.3: Path computation sequence diagram for H-PCE in NCP+.

path according to its own higher-level topology. The related child PCEs are asked to compute the candidate edge-to-edge path segments which are then combined into the resulting end-to-end path and returned to the ingress child PCE which notifies the PCC of the NIPS Server. There is an observation to be made here: the ingress child PCE initially computes paths to all the border nodes. The motivation is that the parent PCE computes a path with the first node on that path always being a border node, without any connectivity information on reaching that border node from the requesting source node. Hence, the child PCE provides this information in its request to the parent PCE, which in turn is able to choose its first border node based on correct intra-domain availability information.

6.4.1 Topology abstraction

The parent PCE has a aggregated or abstracted view of the topology, without any specifics, of each of its child domains. This has two main reasons: (i) confidentiality, as by aggregating, the actual topology of the domains is hidden from the other domains and (ii) scalability, because the end-to-end path computation problem is decomposed and solved over a reduced number of nodes.

In this paper we will discuss and compare two proposals for aggregation schemes: Full Mesh (*FM*) and *Star* aggregation. These aggregation policies are enhanced versions of existing network topology abstractions [19, 28] that only consider network resources: we also incorporate representation of IT resources. In what follows we describe these aggregation techniques and show how to extend them in order for the parent PCE to perform anycast path computations.

In the following section we consider a WDM network $G = (V, E, DC)$ where

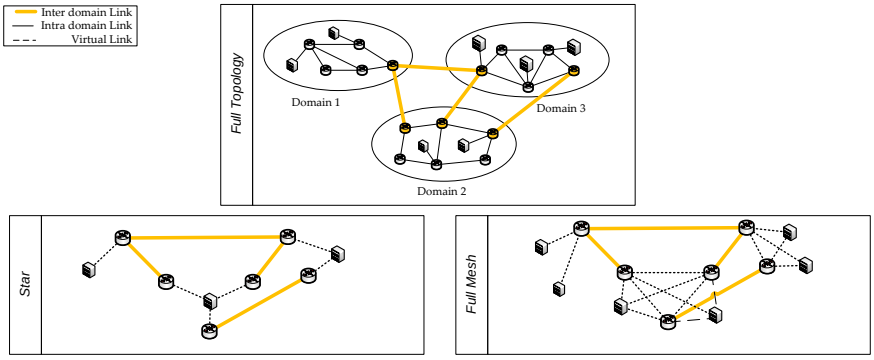


Figure 6.4: The different abstraction methods.

V is the set of nodes, E the set of edges and DC the set of IT-S. The network is divided into D domains $G^i = \left(V^i, E^i, DC^i\right)$ where domain G^i comprises $\left|V^i\right|$ nodes, $\left|E^i\right|$ edges and $\left|DC^i\right|$ IT end points. For each domain we also denote $B^i \subseteq V^i$ as the set of border nodes.

6.4.1.1 Full Mesh abstraction

The aggregated FM topology (see Fig. 6.4) is a transformation of the original topology where the i^{th} domain $G^i = \left(V^i, E^i, DC^i\right)$ is transformed into a graph $G^i_* = \left(V^i_*, E^i_*, DC^i_*\right)$ containing the border nodes $b \in B^i$ and the IT end points $dc \in DC^i$. For every node pair in that subgraph, we create a virtual edge i.e., $\forall k, l \in V^i_* : k \neq l$ create virtual edge $e(k, l)$. The inter-domain links are then copied into the aggregated topology. The virtual links are assigned a cost, computed by the child PCE responsible for that domain. The child PCE computes a path p between all node pairs and calculates its physical length together with a wavelength availability bitmap which indicates if wavelength λ is available on all links along path p. The bitmap ω_p for a path p is computed in eq. 6.1 (b_l is the wavelength bitmap for link l).

$$\omega_p = \bigwedge_{l \in p} b_l \tag{6.1}$$

This information is sent in a Label State Update (LSU) message from the child PCE to the parent PCE (as part of OSPF-TE). As IT resources are not abstracted, IT information is copied and sent from the child PCE to the parent PCE using a PCEP notify message.

The advantage of this aggregation mechanism is that we have a fairly accurate view of the topology and consequently we can compute paths using this detailed wavelength availability. Conversely, the topology can become quite big ($n^2 - n$

links per domain comprising n border nodes and IT resources), which limits the scalability advantage of using a H-PCE method and increases the time needed for path computation.

6.4.1.2 *Star* abstraction

When transforming a domain into a *Star* topology, as depicted in Fig. 6.4, a graph is created that comprises all the border nodes connected to a single virtual node, i.e. $\forall G^i = \left(V^i, E^i, DC^i\right)$ create n^i_* and connect it to every $b \in B^i$. This n^i_* will not only serve as a connection point for the border nodes, but is also the representation of an aggregated IT endpoint in that domain. All inter-domain links are then copied into the topology.

Assigning link weights for the *Star* aggregation is more complex than for ***FM***, as all possible paths between a certain pair of border or IT nodes, are abstracted into a single two-link path in the *Star* topology. We will compare three approaches to compute the availability ω_l (number of available wavelengths) and a length metric $|l|$ for each virtual link l with one of the border nodes as a source. (Links with the virtual node as an end point have no special meaning and are always available with length metric 0).

1. *Bin*ary: each virtual link in the aggregated topology receives a binary availability and unit length. The link is only updated to unavailable when no border node and no IT-S can be reached any more from that border source node (due to a lack of network resources).

$$\omega_l = \begin{cases} 1 & \text{if available} \\ \infty & \text{if unavailable} \end{cases} \tag{6.2a}$$

$$|l| = 1 \tag{6.2b}$$

2. Avg: for each border node $b \in B_i$, we calculcate a set S_b of intra-domain paths to every other border domain node and IT-S. The number of available wavelengths for the virtual link l connecting b with the central node is then calculated as the average number of free wavelengths from the availability maps from the calculated paths (computed as in Eq. 6.1) - see Eq. 6.3a while the length is the average of the path lengths from the paths $p \in S_b$ - see Eq. 6.3b. (We denote the number of available wavelengths in a bitmap ω_p as ω^λ_p and the length of path p as $|p|$).

Table 6.3: Routing metric used by the shortest path algorithm in the parent and child PCE modules.

Name	Description
SP	Physical length Shortest Path routing: $\pi(l) = \|l\|$
AV	For each link l use its current fraction of used wavelengths: $\pi(l) = \frac{\omega_l}{\omega_{total}}$ where ω_{total} represents the total number of wavelengths of link l and ω_l the number of active wavelengths. AV forces links with a higher load to be less likely used.
AV-L	For each link l we use $\pi(l) = \|l\| \times \frac{\omega_l}{\omega_{total}}$. This way, the algorithm favours shorter paths with high availability.

$$\omega_l = \frac{1}{|S_b|} \cdot \sum_{p \in S_b} \omega_p^\lambda \tag{6.3a}$$

$$|l| = \frac{1}{|S_b|} \cdot \sum_{p \in S_b} |p| \tag{6.3b}$$

3. Max: for each link l, connecting border node b with the central node, we calculate the same set S_b as for Avg, but now ω_l equals ω_p^λ for path $p \in S_b$ with the highest number of available wavelengths (see Eq. 6.4a), while the length $|l|$ corresponds to the actual length of that respective path (see Eq. 6.4b).

$$\omega_l = \max_{p \in S_b} \left(\omega_p^\lambda\right) \tag{6.4a}$$

$$|l| = |p| : \max_{p \in S_b} \left(\omega_p^\lambda\right) \tag{6.4b}$$

With *Star* abstraction, the resulting aggregated topology is considerably smaller ($2.n$ if n is the number of border nodes), improving the scalability of H-PCE and minimizing the path computation time at the parent PCE. Moreover, employing Bin limits the label state updates (LSU) to updates for the inter-domain links. However, the drawback is that computed paths may be suboptimal, leading to a potentially higher blocking ratio.

6.4.2 Routing algorithms

In this paper we consider three metrics, described in Table 6.3, used by a shortest path routing algorithm (e.g., Dijkstra), employed by the PCE child and parent modules. (We denote the weight for link l as $\pi(l)$).

Table 6.4: Scheduling Mechanisms.

Symbol	Description
L-max	Schedule to the IT-S with the highest current load, concentrating requests at the same location as much as possible. This algorithm is used, e.g., when IT energy minimization is of concern [36].
L-Min	Choose the IT-S with the lowest current load, performing IT load balancing.
Closest	We schedule to the IT-S which is closest in terms of the employed link metrics (SP, AV or AV-L).
Random	We randomly select an IT-S for benchmarking purposes.

6.4.3 Scheduling algorithms

We consider a number of IT-S scheduling strategies listed in Table 6.4, which we apply for *Star* and *FM* aggregation. These scheduling mechanisms use the exact information of the IT-S for the *FM* abstraction. *Star* however, aggregates a domain's IT information into one virtual node, for which we average all the information per domain (current processing load, maximum available capacity, etc.) and send it to the PCE child.

6.5 Simulation results

The performance of the different aggregation mechanisms and routing algorithms is evaluated by simulation. We have built a simulation environment based on OMNeT[2], which is fully described in [38]. A 9 domain, 72 node optical network has been considered, with 18 data centers as shown in Fig. 6.5. There are 38 border nodes and 96 bidirectional links (of which 24 inter domain links). Each intra-domain and inter-domain link accommodates 32 and 64 wavelengths respectively and we consider a network with wavelength conversion. Hence, the effect of resource fragmentation on blocking in our use case is not present, as blocking only occurs when there is no more free network or IT capacity. Each data center has 500 servers. A request originating at a source site asks for a certain amount of servers that need to be reserved at a destination site of choice, and one unit of bandwidth (i.e., a wavelength) between the source site and the chosen destination site. In order to accommodate a request, a destination site needs to be chosen with enough available capacity and a path needs to be computed between the source site and the destination site. The numbers shown in the graphs are averages of 20 simulations with a different seed. The 95% confidence intervals are very small, so

[2]http://www.omnetpp.org/

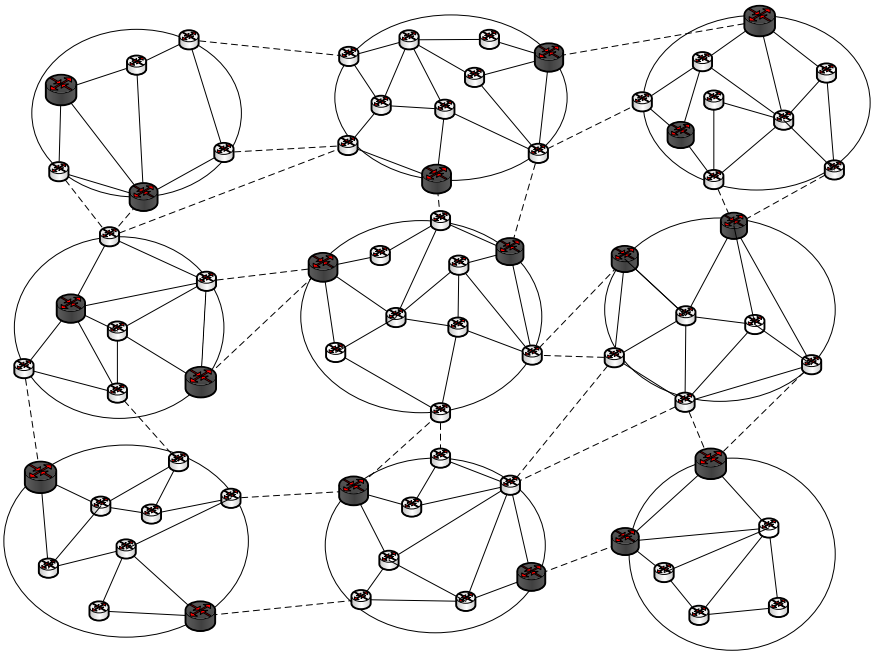


Figure 6.5: The topology considered for the simulations, taken from [37]. The dark grey nodes represent the nodes that have an attached IT-S.

we have opted not to draw them to make the graphs more clear. We stopped simulation after 200.000 requests have been processed. In our simulations, the requests are generated using a Poisson process (with exponentially distributed arrival and service rate) and are scheduled following one of the scheduling mechanisms described in Section 6.4.3. We apply a uniform traffic profile, where each node in every domain has the same arrival rate. We only choose among destination sites that can accommodate the requested capacity; if there are multiple equivalent IT-S (e.g., *L-max* where two IT-S have the same load), we choose the IT-S which is closest (in terms of the metrics used by the routing algorithm).

We have divided the results in two main categories: (i) the *network-intensive scenario* where there is always enough IT capacity (i.e., each request only requires one server), thus blocking only occurs because of lack of network resources (called network blocking) and (ii) the *computing-intensive scenario* where the number of requested servers is significant (15 servers), hence blocking occurs because of either lack of network or IT resources (the latter is called IT blocking).

We first investigate the different aggregation schemes separately, trying to find the best

- scheduling technique (Section 6.5.1),

- routing algorithm (Section 6.5.2),
- and network information abstraction methods (Section 6.5.3)

for both *Star* and *FM*. Subsequently we compare these *Star* and *FM* strategies on service blocking, end-to-end setup time and network control plane load in Section 6.5.4.

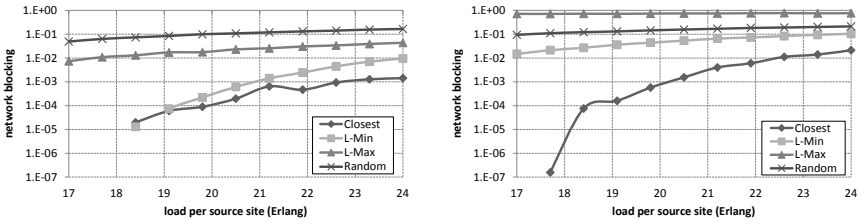
We do this for:

- the *network-intensive scenario*
- the *computing-intensive scenario*

6.5.1 Scheduling algorithm (*network-intensive scenario*)

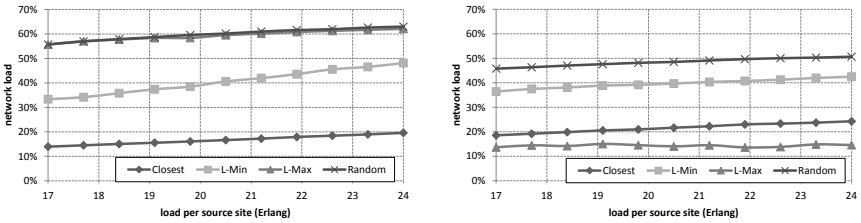
We want to find the best scheduling algorithm terms of service blocking. Fig. 6.7a and Fig. 6.7b show this blocking for *FM* and *Star* aggregation respectively. First, we notice that the best scheduling policy (for both *FM* and *Star*) is *Closest* as expected: we schedule to the nearest IT-S minimizing the required number of network resources and as there is always enough IT capacity, blocking due to a lack of IT resources never occurs. Secondly, we notice that either IT load balancing (*L-Min*) or concentrating requests in one location (*L-max*), leads to a significant service degradation. This is also reflected in Fig. 6.7c and Fig. 6.7d, which show the average network load (defined as the ratio of the number of active and the total number of wavelengths). *Closest* requires the least amount of network resources (as shorter paths are chosen) and *L-Min* and *L-max* require about 24% and 43% more network resources in the *FM* case. We note that these qualitative conclusions remain (*Closest* is always best) for all routing algorithms and network abstraction methods.

Lastly we note that for *Star* aggregation, *L-max* has substantially high network blocking values. The reason is that *Star* aggregation uses abstracted information for its virtual links. Initially, a domain is able to accommodate requests and hence one IT-S is chosen to do this. As more and more requests are scheduled to that IT-S, there is a point where no path can be found between one of the domain's border nodes and that IT-S. So, although still having enough IT capacity, the IT-S is unable to serve any more request as it has become unreachable in the real topology, while the abstracted topology tells otherwise. The metrics used for the *Star* abstraction cannot reflect this unavailability as one virtual link abstracts multiple paths. Hence, the scheduling mechanism chooses a reachable IT-S in the abstracted topology, which is unreachable in the actual topology and the request is blocked when an intra-domain path needs to be found by the child PCE responsable. As no requests are provisioned, the network load is also low (see Fig. 6.7d). We note that this effect also happens for the *computing-intensive scenario* and have chosen not to show the results for *Star- L-max* for the *computing-intensive scenario*.



(a) Network blocking for *FM*.

(b) Network blocking for *Star*.



(c) Network load for *FM*.

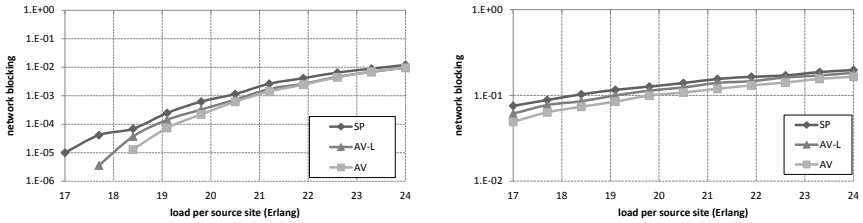
(d) Network load for *FM*.

Figure 6.6: Network blocking and network load figures for Star and FM aggregation, using AV as routing algorithm and Avg as information abstraction method for Star for the network-intensive scenario. Closest scheduling minimizes network blocking and network resource load. The relative position of the scheduling mechanisms remain the same for routing algorithms or network abstraction methods.

6.5.2 Routing algorithms (*network-intensive scenario*)

6.5.2.1 *FM* Routing algorithms

The impact of routing algorithms on service blocking depends on the scheduling strategy. For scheduling strategies where the destination site remains constant for a certain period (i.e., *Closest* and *L-max*), the difference in service blocking among varying routing algorithms is minor, which means that optimal routing amounts to the choice of a shortest path. We do not show the blocking figures, because of space limitations. For the scheduling strategies where the destination choice varies more over time (i.e., *L-Min* and *Random*), we notice a subtle distinction between routing algorithms. Fig. 6.8a and Fig. 6.8b show that the network load balancing algorithm (AV routing) minimizes blocking, closely followed by AV-L and SP. Consequently, when the choice of destination IT-S is more dynamic during a certain period, the wavelength availability information is exploited to find better paths, lowering the network blocking.



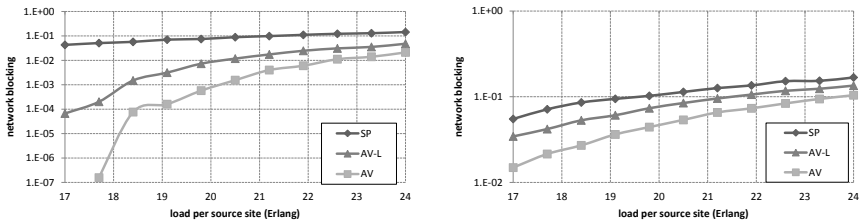
(a) *L-Min* (b) *Random*.

Figure 6.7: Network blocking figures for L-Min and Random for FM abstraction for the network-intensive scenario. We can see that AV routing is preferred, but that differences are subtle.

6.5.2.2 *Star* routing algorithms

The distinction between routing algorithms becomes apparent when applied in *Star* aggregation. We show the blocking in Fig. 6.9a and Fig. 6.9b for *Closest* and *L-Min* respectively using Avg information abstraction (but conclusions also apply for *Random* and *L-max* and other information abstraction methods). Incorporating the aggregated network availability information per border domain node into the routing algorithm (i.e., AV and AV-L routing) optimizes the network blocking: (i) the inter-domain chain can be chosen according to the abstracted wavelength availability information and (ii) the intra-domain paths can be optimized using the exact wavelength availability information.

We also note that AV-L never outperforms AV: the wavelength availability information suffices to find the optimal choice for paths, while the information on the exact distance of the paths does not play a crucial role.



(a) *Closest* (b) *Random*.

Figure 6.8: Network blocking figures for Closest and L-Min scheduling for Star abstraction (using Avg as information aggregation) for the network-intensive scenario. The differences between routing algorithms are clearly defined: AV has the best performance, followed by AV-L and SP.

6.5.3 Information aggregation (*network-intensive scenario*)

In Fig. 6.9 we compare network blocking figures between three methods to abstract the domain's information : (i) ***Bin***, (ii) Avg and (iii) Max. Note that we only include the graph for AV routing with *Closest* scheduling, but conclusions qualitatively apply for the other routing/scheduling strategies. We see that the best way to abstract the information, is averaging the network information. Max is unable to achieve the same network blocking as Avg: choosing information from one representative path as abstraction method cannot attain the same service blocking as an aggregated representation of the whole domain.

We also note that *Closest* scheduling with ***Bin*** as abstraction method, leads to pure intra-domain scheduling and routing: all requests are scheduled to one of the available IT-S in the same domain. The distance from the requesting source to the domain's virtual node (abstract IT-S) is either one or unavailable. Distances to another domain's virtual node would require a distance larger than one, and are consequenlty never chosen. This observation confirms the need for inter-domain scheduling: a sheduling and routing strategy which is able to perform inter-domain scheduling and routing (e.g., AV-*Closest* with Avg as abstraction schedules 62% intra-domain and 38% inter-domain), outperforms in terms of blocking every possible routing and scheduling strategy which schedules requests only intra-domain.

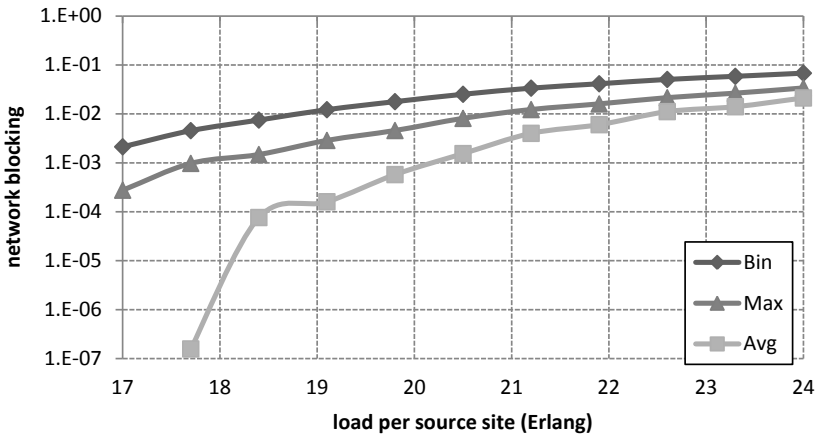


Figure 6.9: Comparison of network blocking for the different network information abstraction methods for Star abstraction with, Closest scheduling and AV routing.

6.5.4 *FM* vs *Star* for the *network-intensive scenario*

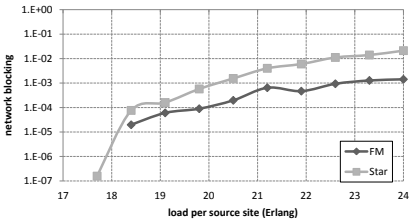
We compare performance metrics for *FM* with those from *Star* aggregation. Above, we have concluded that AV routing together with *Closest* scheduling achieves minimum service blocking (together with Avg abstraction for *Star*). Hence we compare those strategies on service blocking, end-to-end setup time and network control plane load in Fig. 6.10. The setup time has three contributions:

1. Time required to exchange control messages.
2. Time required to compute the path
3. Time required to configure the optical devices. For this we have used 50ms, which is the worst case scenario for micro-eletromechanical system (MEMS) cross-connects configurations [39].

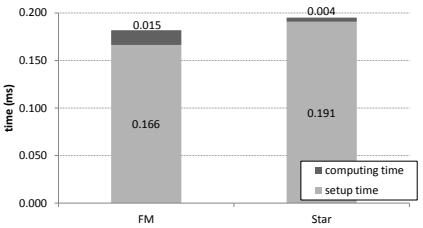
As expected, Fig. 6.11a shows that the increased connectivity information for each border node seems to be beneficial as *FM* has a lower blocking probability than *Star* (between 2 and 14 times smaller). The time to compute a path however, is much smaller for *Star* then *FM*, which can be seen in Fig. 6.11b: *Star* computes its paths about 3.5 times faster. However, *Star* computes suboptimal paths which are about 10% longer than the paths computed by *FM*. Hence, the decrease in computation time is outweighed by the extra time needed to set up the longer path: *FM* is able to set up a path in about 6% less time. In addition, these longer paths also increase the number of the OSPF Label State Updates (LSU) that need to be exchanged, which increases complexity for the network control plane: on average 24% more LSU messages are needed (see Fig. 6.11d). Concluding, (i) the increased service blocking ratio, (ii) the longer paths which are computed, (iii) the associated longer setup times and, (iv) the increased network control plane load turn the *Star* aggregation as a redundant technique for a *network-intensive scenario*. The scalability motivation (reduced number of virtual links in the aggregated topology) is nullified as the time needed to set up a path is still larger than the more complex *FM* technique.

6.5.5 Computing-intensive scenario

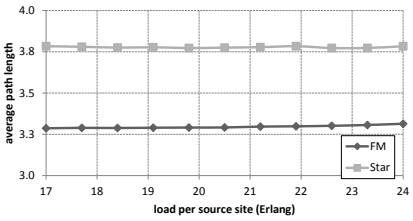
In the *computing-intensive scenario*, we increase the number of servers demanded per request from 1 to 15. Here, situations may occur where there is not enough IT capacity and hence both network and IT blocking may occur. Our simulations point out that some of the conclusions drawn for the *network-intensive scenario*, also apply here: (i) for both *Star* and *FM*, AV routing is the best routing mechanism, (ii) *Closest* is still the best scheduling strategy for *Star*, (iii) Avg information abstraction for *Star* is still the best strategy and (iv) inter-domain schedul-



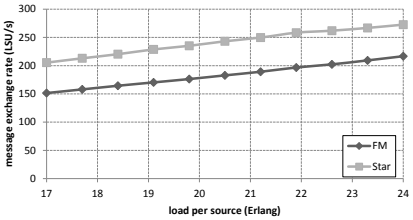
(a) Network blocking between for *FM*/ *Star*.



(b) End-to-end setup and path computation time for *FM*/ *Star*.



(c) Average path length for *FM*/ *Star*.



(d) LSU message exchange rate between the parent PCE and child PCE modules for *FM*/ *Star*.

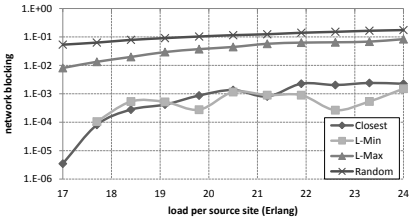
Figure 6.10: This figure compares the network blocking, end-to-end setup times, average path length and the average number of LSU messages exchanged between the parent PCE and its children for AV-Closest (and Avg information abstraction for Star). FM leads to lower network blocking, shorter paths, lower end-to-end setup times and a reduction in network control plane load for the network-intensive scenario.

ing shows to decrease service blocking compared to a scenario where only intra-domain scheduling is performed.

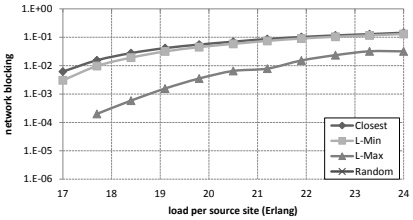
However, the relation between scheduling strategies changes for *FM*, which can be observed in Fig. 6.11, where we show the amount of blocking due to insufficient network (network blocking, Fig. 6.12a) and IT resources (IT blocking, Fig. 6.12b). The sum is the total blocking, Fig. 6.12c.

We see in Fig. 6.12c that up to 18.5 Erlang, *L-Min* achieves lowest total blocking, but for higher loads *L-max* is best. This is attributed to the fact, that in a low load scenario *L-max* computes longer paths to distant IT-S, leading to a higher network blocking ratio (see Fig. 6.12a). Hence, *L-max* achieves a lower computational resource load than *L-Min* and *Closest* (requests are blocked) and consequently, the moment where *L-Min* and *L-max* attain the same IT resource load is different (e.g., *L-max* reaches an average IT resource load of 87% at 24 Erlang, while *L-Min* reaches this at 20 Erlang). Since in higher load scenarios, *L-max* has more available IT capacity, it attains lower IT blocking and because the contribution of IT blocking to the total blocking is considerable, it also attains lower total

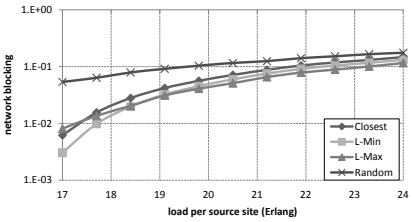
blocking.



(a) Network blocking.



(b) IT Blocking.



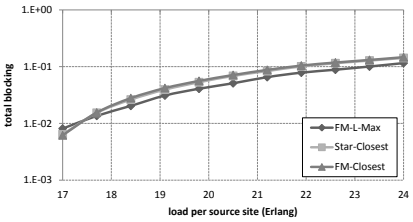
(c) Total blocking.

Figure 6.11: The network, IT and total blocking for FM with AV routing. Although Closest and L-Min achieve the minimal network blocking, IT blocking is relatively high. L-max is able to reduce the IT blocking penalty, which is reflected in the total blocking figures.

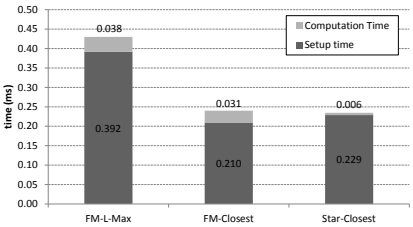
When comparing *Star* and *FM* in terms of service blocking in Fig. 6.13a, we see that *Star* with *Closest* scheduling has about the same service blocking figures as *FM* with *Closest* scheduling. Indeed, *Closest* scheduling is not the best strategy for *FM* (as it achieves high IT blocking values). When comparing *Star-Closest* (best choice for *Star* aggregation) with *FM* with *L-max* scheduling however, we see that *FM* is again able to lower its service blocking ratio noticeably. Nevertheless, this intelligent scheduling of *FM* comes at a price, as shown in Fig. 6.12. To achieve the decrease in service blocking, *FM- L-max* computes longer paths (as shown in Fig. 6.13c) which comes at two extra costs: (i) a longer setup time (longer paths) for *FM- L-max* with higher computing time leads to a higher end-to-end setup time (see Fig. 6.13b) and (ii) more network control plane load as more LSU messages need to be sent to the parent PCE module.

Comparing *FM- Closest* with *Star- Closest*, we observe that they attain about the same blocking ratio. *FM* however, is able to reduce the path length compared to *Star* which balances out the time required to compute the path: when enforing *Closest* scheduling *Star* has only 2.21 % faster end-to-end setup times. Consequently, when fast setup times are required, the operator has two options: (1) run *FM- L-max* and *Star- Closest* in parallel and choose which abstraction method to

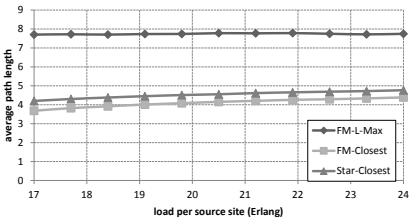
choose based on the setup time requirements or (2) run *FM- L-max* and schedule using the *Closest* strategy when the setup time requirements are important.



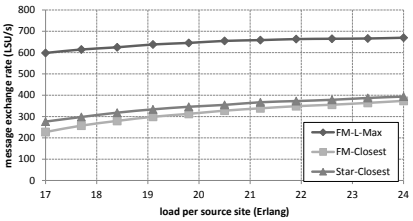
(a) Comparing network blocking between *FM* and *Star*, for *Closest* and *L-max*.



(b) Comparing the end-to-end setup time for *Star* and *FM*.



(c) Comparing the path lengths for *Star* and *FM*.



(d) Amount of LSU messages exchanged between the parent and child PCE modules, for *Star* and *FM*.

Figure 6.12: Comparing FM and Star on total blocking, end-to-end setup time, average path length and network control plane load with Closest scheduling, AV routing and Avg information for the computing-intensive scenario. FM still achieves lower total blocking, but cannot achieve lower end-to-end setup times then Star.

6.6 Conclusion

Today, we observe an evolution to network-based service offerings, where applications are pushed further into the network and increasingly rely on interworking of many distributed components: the cloud computing paradigm. Given the increasing adoption of the cloud ideas (cf. IaaS, PaaS, SaaS), in both the consumer, business and even academic spaces, service providers are confronted with more stringent and/or demanding network requirements (in terms of bandwidth, latency) as well as the need to incorporate IT resources in their offering. Therefore, it becomes essential that the service management system is able to manage both network and IT resources in a coordinated way. This paper proposes a set of extensions to the well know Generalized Multi-Protocol Label Switching (GMPLS) and Path Computation Element (PCE)-based Network Control Plane for an optical,

multi-domain routing scenario. The NCP+ is aware of the data centers attached to its network and is able to compute anycast paths to one of these IT end points. Our proposed NCP+ architecture provides protocol extensions to disseminate IT information, and includes enhanced topology information aggregation schemes and joint network and IT resource selection and allocation policies. We have evaluated these schemes and policies using simulation with general conclusions summarized in Table 6.5. We have demonstrated that for a scenario where applications have very strict network but flexible IT requirements, *FM* abstraction performs best in terms of service blocking, end-to-end setup times and added network control plane load. However, in a scenario where IT requirements are dominant, running both algorithms in parallel could lead to an improvement in service blocking and end-to-end setup time. Future work includes an investigation in an adaptive approach: either using *Star* or *FM* depending on the IT vs. network load.

Table 6.5: This table summarizes the best scheduling, routing and information abstraction techniques per aggregation method and scenario in terms of service blocking. The last line sums up the conclusions for the comparison between FM and Star per scenario.

	network-intensive scenario	***computing-intensive scenario***
Star	• *Closest* scheduling • AV routing • AVG abstraction	• *Closest* scheduling • AV routing • AVG abstraction
FM	• *Closest* scheduling • AV/SP routing	• *L-max* scheduling • AV routing
Star vs. *FM*	• *FM* computes better paths in terms of service blocking • Due to suboptimal paths, the reduction in computation time of *Star* is nullified and *FM* has faster setup times.	• *FM- L-max* reduces service blocking the most • *FM- L-max* has higher setup times than *Star Closest* and *FM- Closest*, which correspond in service blocking and setup time.

References

[1] C. Develder, M. De Leenheer, B. Dhoedt, M. Pickavet, D. Colle, F. De Turck, and P. Demeester. *Optical networks for grid and cloud computing applications*. Proc. IEEE, 100(4):1149–1167, Apr. 2012.

[2] H. Urs and B. Luiz Andre. *The datacenter as a computer: an introduction to the design of warehouse-scale machines*, volume 6 of *Synthesis lectures in computer architecture*. Morgan and Claypool, 2009.

[3] D. Verchere. *Cloud computing over telecom networks*. In Proc. of. Optical Fiber Commun. Conf. and Exposition and the Nat. Fiber Optic Engineers Conf. (OFC/NFOEC), pages 1–3, Los Angeles, USA, 6–10 March 2011.

[4] L. Garber. *Converged infrastructure: Addressing the efficiency challenge*. IEEE Computer, 45(8):17–20, aug. 2012.

[5] J. Buysse, M. De Leenheer, B. Dhoedt, and C. Develder. *Providing resiliency for optical grids by exploiting relocation: A dimensioning study based on ILP*. Comput. Commun., 34(12):1389–1398, Aug. 2011.

[6] K. Walkowiak. *Anycasting in connection-oriented computer networks: models, algorithms and results*. Int. J. of Applied Math. and Comp. Science, 20(1):207–220, 2010.

[7] B. Bathula, J. Plante, and V. Vokkarane. *Crosstalk-aware anycast routing and wavelength assignment in optical WDM networks*. In Proc. IEEE 4th Int. Symp. on Advanced Netw. and Telecom. Systems (ANTS), pages 94–96, Mumbai, India, 16–18 Dec. 2010.

[8] J. Buysse, C. Cavdar, M. De Leenheer, B. Dhoedt, and C. Develder. *Improving energy efficiency in optical cloud networks by exploiting anycast routing*. In Proc. of Asia Commun. and Photonics Conf., volume 8310, pages 1–6, Shangai, China, 13–16 Nov. 2011.

[9] J. Buysse, K. Georgakilas, A. Tzanakaki, M. De Leenheer, B. Dhoedt, C. Develder, and P. Demeester. *Calculating the minimum bounds of energy consumption for cloud networks*. In Proc. of IEEE Int. Conf. Comp. Commun. and Networks (ICCCN), pages 1–7, Maui, USA, 31Jul. – 4 Aug. 2011.

[10] K. Bhaskaran, J. Triay, and V. Vokkarane. *Dynamic anycast routing and wavelength assignment in WDM networks using ant colony optimization (ACO)*. In Proc. IEEE Int. Conf. on Commun. (ICC), pages 1–6, Kyoto, Japan, 5–9 Jun. 2011.

[11] E. Mannie. *IETF, RFC 3945 : generalized multi-protocol label switching (GMPLS) architecture*. Technical report, Network Working Group, Oct. 2004.

[12] A. Farrel, J.-P. Vasseur, and J. Ash. *IETF, RFC 4655: a path computation element PCE-based architecture*. Technical report, Network Working Group, Aug. 2006.

[13] S. Figuerola, N. Ciulli, M. De Leenheerc, Y. Demchenko, W. Zieglere, and A. Binczewski. *PHOSPHORUS: Single-step on-demand services across multi-domain networks for e-science*. In Proc. European Conf. and Exhibition on Optical Commun. (ECOC), pages 1–16, Berlin, Germany, 16–22 Sep. 2007.

[14] T. Metsch. *Open cloud computing interface - use cases and requirements for a cloud API*. Technical report, Open Grid Forum, 16 Sep. 2009.

[15] S. Andreozzi, M. Sgaravatto, and M. C. Vistoli. *Sharing a conceptual model of grid resources and services*. Comp. Resource Repo., pages 1–4, 18 Jun. 2003.

[16] J. MacLaren. *HARC: the highly-available resource co-allocator*. In R. Meersman and Z. Tari, editors, On the Move to Meaningful Internet Systems 2007: CoopIS, DOA, ODBASE, GADA, and IS, volume 4804 of *Lecture Notes in Computer Science*, pages 1385–1402. Springer Berlin Heidelberg, 2007.

[17] J. MacLaren. *Co-allocation of Compute and Network resources using HARC*. In Proc. Lighting the Blue Touchpaper for UK e-Science: Closing Conf. of the ESLEA Project, pages 1–5, Edinburgh, UK, 26–28 Mar. 2007.

[18] S. R. Thorpe, L. Battestilli, G. Karmous-Edwards, A. Hutanu, J. MacLaren, J. Mambretti, J. H. Moore, K. S. Sundar, Y. Xin, A. Takefusa, M. Hayashi, A. Hirano, S. Okamoto, T. Kudoh, T. Miyamoto, Y. Tsukishima, T. Otani, H. Nakada, H. Tanaka, A. Taniguchi, Y. Sameshima, and M. Jinno. *G-lambda and EnLIGHTened: wrapped in middleware co-allocating compute and network resources across Japan and the US*. In Proc. Int. Conf. on Netw. for grid applications, pages 5:1–5:8, Lyon, France, 17–19 Oct. 2007.

[19] M. Chamania and A. Jukan. *A survey of inter-domain peering and provisioning solutions for the next generation optical networks*. IEEE Commun. Surveys and Tutorials 11, 1:33–51, 2009.

[20] J. Vasseur, A. Ayyangar, and R. a. Zhang. *IETF, RFC 5152 : a per-domain path computation method for establishing inter-domain traffic engineering*

(TE) label switched paths (LSPs). Technical report, Networking Working Group, Feb. 2008.

[21] A. Farrel, A. Satyanarayana, A. Iwata, N. Fujita, and G. Ash. *IETF, RFC 4920 - Crankback Signaling Extensions for MPLS and GMPLS RSV*. Technical report, Network Working Group, Jul. 2007.

[22] J. Vasseur, R. Zhang, N. Bitarn, and J. Le Roux. *IETF, RFC 5441: a backward-recursive PCE-based computation procedure to compute shortest constrained inter-domain traffic engineering label switched paths.* Technical report, Network Working Group, 9 Apr. 2009.

[23] S. Dasgupta, J. de Oliveira, and J.-P. Vasseur. *Path computation element based architecture for interdomain MPLS/GMPLS traffic engineering: Overview and performance*. Netw. Arch., Manag., and App., 21:38–45, 2007.

[24] S. Sanchez-Lopez, J. Sole-Pareta, J. Comellas, J. Soldatos, G. Kylafas, and M. Jaeger. *PNNI-based control plane for automatically switched optical networks*. IEEE J. of Lightwave Techn., 21:2673–2682, Nov. 2003.

[25] L. Contreras, V. Lopez, O. De Dios, A. Tovar, F. Munoz, A. Azanon, J. Fernandez-Palacios, and J. Folgueira. *Toward cloud-ready transport networks*. IEEE Commun. Mag., 50(9):48–55, Sep. 2012.

[26] J. Triay, S. G. Zervas, C. Cervelló-Pastor, and D. Simeonidou. *GMPLS/PCE/OBST architectures for guaranteed sub-wavelength mesh metro network services*. In Optical Fiber Commun. Conf. (OFC), pages 1–3, Los Angeles, USA, 6–10 Mar. 2011.

[27] F. Paolucci, O. G. de Dios, R. Casellas, S. Duhovnikov, P. Castoldi, R. Munoz, and R. Martinez. *Experimenting hierarchical PCE architecture in a distributed multi-platform control plane testbed*. In Proc. Optical Fiber Commun. Conf. (OFC), pages 1–3, Los Angeles, USA, 4–8 Mar. 2012.

[28] S. Uludag, K.-S. Lui, K. Nahrstedt, and G. Brewster. *Analysis of topology aggregation techniques for QoS routing*. ACM Comput. Surv., 39(3):7–38, Sep. 2007.

[29] Q. Liu, M. Kok, N. Ghani, V. Muthalaly, and M. Wang. *Inter-domain provisioning in DWDM networks*. In Proc. IEEE Global Telecommun. Conf. (GLOBECOM), pages 1–6, San Francisco, CA, USA, 27 Nov.–1 Dec. 2006. doi:10.1109/GLOCOM.2006.401.

[30] G. Maier, C. Busca, and A. Pattavina. *Multi-domain routing techniques with topology aggregation in ASON networks*. In Proc. Int. Conf. on Optical Netw.

Design and Modeling (ONDM), pages 1–6, Vilanova i la Geltru, Spain, Mar. 2008.

[31] D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, and G. Swallow. *IETF, RFC 3209 : RSVP-TE: Extensions to RSVP for LSP Tunnels*. Technical report, Network Working Group, Dec. 2001.

[32] J. Moy. *IETF, RFC 2328: OSPF Version 2*. Technical report, Network Working Group, Apr. 1998.

[33] S. Shew and J. Sadler. *User Network Interface (UNI) 2.0 Signaling Specification*. Technical report, Optical Internetworking Forum, Feb. 2008.

[34] J. Vasseur and J. L. Le Roux. *IETF, RFC 5440 :path computation element (PCE) communication protocol (PCEP)*. Technical report, Network Working Group, Mar. 2009.

[35] C. Ciulli, G. Landi, F. Salvestrini, D. Parniewicz, X. Chen, G. Buffa, P. Donadio, Y. Demchenko, C. Ngo, J. Jimenez, A. Tovar De Duenas, C. Garcia Argos, P. Robinson, A. Manfredi, P. Drozda, Brzozowski, J. Ferrer Riera, S. Spadaro, E. Lopez, E. Escalona, M. Antoniak-Lewandowska, L. Drzewiecki, A. Tzanakaki, K. Georgakilas, and J. Buysse. *Deliverable D4.1 GMPLS+/PCE+ control plane architecture*. Technical report, GEYSERS Project, Nov. 2010. Available from: http://www.geysers.eu/images/stories/deliverables/geysers-deliverable_4.1.pdf.

[36] J. Buysse, K. Georgakilas, M. De Leenheer, B. Dhoedt, and C. Develder. *Energy-Efficient resource provisioning algorithms for optical clouds*. accepted for publication in to J. of Opt. Commun. and Netw. (JOCN), July 2012.

[37] S. Shang, N. Hua, L. Wang, R. Lu, X. Zheng, and H. Zhang. *A hierarchical path computation element (PCE)-based k-random-paths routing algorithm in multi-domain WDM networks*. Opt. Switch. and Netw., 8(4):235–241, 2011.

[38] M. De Leenheer, J. Buysse, K. Mets, B. Dhoedt, and C. Develder. *Design and implementation of a simulation environment for network virtualization*. In Proc. 16th IEEE Int. Workshop Comp. Aided Modeling, Analysis and Design of Commun. Links and Netw. (CAMAD), pages 87–91, Kyoto, Japan, 10–11 Jun. 2011.

[39] A. Olkhovets, P. Phanaphat, N. C., D. Shin, C. Lichtenwalner, M. Kozhevnikov, and J. Kim. *Performance of an optical switch based on 3-D MEMS crossconnect*. IEEE Photonics Techn. Letters, 16(3):780–782, March 2004. doi:10.1109/LPT.2004.823703.

7
Conclusions

"Good morning," said Deep Thought at last.
"Er good morning, O Deep Thought," said Loonquawl nervously, "do you have . . . er, that is . . . "
"An answer for you?" interrupted Deep Thought majestically. "Yes. I have."
The two men shivered with expectancy. Their waiting had not been in vain.
"There really is one?" breathed Phouchg.
"There really is one." confirmed Deep Thought.
"To Everything? To the great Question of Life, the Universe and Everything?"
"Yes."
. . .
Though I dont think," added Deep Thought, "that you are going to like it."
. . .
"Tell us!"
"All right," said Deep Thought. "The Answer to the Great Question . . . "
"Yes!"
"Of Life, the Universe and Everything . . . " said Deep Thought.
"Yes!"
"Is" said Deep Thought, and paused.
"Yes!"
"Is"
Yes!!! . . . ?"
"Forty-two," said Deep Thought, with infinite majesty and calm.
"Forty-two?"

"I think the problem such as it was, was too broadly based. You never actually stated what the question was."
"B- b- but it was the Ultimate question, the question of Life, the Universe, and Everything."
"Exactly. Now that you know that the answer to the Ultimate question of Life, the Universe, and Everything is forty-two, all you need to do now is find out what the Ultimate Question is."

– Douglas Adams, The Hitchhiker's Guide to the Galaxy

7.1 Main contributions of this work

This work has addressed a set of relevant issues that grid and cloud architectures are currently facing. We initially started by explaining the development of grid and cloud computing, by giving examples of state-of-the-art e-Science, business and consumer applications which are employing these distributed systems. To overcome the present-day confusion regarding the definitions of cloud and grid computing, we formally described them by providing checklists of properties these architectures should possess. From these checklists, it was clear that grid and cloud computing are very closely related to each other as they require a similar coordination of resources. As argued in Chapter 2, optical circuit switching based on WDM technology is a perfect candidate to support current grid and cloud deployments, as it provides cost-effective, high bandwidth connections with low latency. This led us to the definition of the "optical grid" and "optical cloud".

Our work addressed four optical grid/cloud issues. First, we have developed a simulation environment, which is able to put optical grid/cloud algorithms and architecture propositions to the test. Secondly, we addressed the issue of resiliency: how can we efficiently provision network and IT resources, able to overcome single link failures. We continued by investigating the energy expenditure of an integrated network and IT infrastructure, by proposing a provisioning algorithm lowering the infrastructure's energy consumption by allowing switching off unused resources. Lastly, we proposed an enhanced network control plane architecture, the Network Control Plane +, which uses the scalable hierarchical path computation element architecture to compute IT end points and their corresponding connections.

7.1.1 Optical grid/cloud simulation environment

To perform full scale validation, and perform extensive testing of the algorithms and architecture described throughout this book, experiments based on discrete event simulation have been identified as the most appropriate method to study the performance. The blueprints for this simulation environment have been described

in Chapter 3. This simulator consists of three parts: (i) a database containing the static information of the physical infrastructure, (ii) a part which is able to virtualize network and IT resources and (iii) a component able to provision over the virtualized resources. Our layered design has the advantage of (i) freeing up memory for the dynamic parts as scalability is of major concern and (ii) being able to perform simulations focusing on one part of the stack, without the overhead interactions of the other components.

7.1.2 Resiliency in optical grids/clouds

In Chapter 4 we investigated resiliency against network failures with the adoption of shared path protection under the anycast routing principle, for the grid/cloud to survive from any possible single link failure. Traditional protection strategies force the primary and backup paths to have the same end point. We however, allow the primary and backup IT site for a given request to differ, which is achieved by the anycast principle. Our studies have shown that this *shared path protection with relocation*, is able to lower the number of consumed wavelengths considerably (up to 21% for the considered topologies and demand patterns) . The decrease mainly stems from a reduction in backup wavelengths by either relocating to another closer server site or exploiting a sharing possibility which was not possible in the traditional case. The relative amount of saved network resources is dependent on two factors.

- **Degree of the topology:** in a sparse network, a cycle composed by a primary path and its corresponding backup path will be quite long on average. Consequently, providing a backup path to another resource can drastically reduce the length of the route towards that closer backup resource, especially when that new backup resource lies on the original backup path. The effect presents itself less in a topology with a higher degree.
- **Number of IT sites:** using more servers implies a higher probability of encountering another server on the backup path to the original one, and thus relocation is favorable.

We devised three methods for computing the primary and backup IT end points and the routes towards them.

1. An ILP formulation.
2. Two heuristics, H1 and H2.
3. A solution method based on column generation.

The first method is a well known technique, applied in the literature. While this solution method allows to find an optimal answer, it suffers from ILP's inherent

scalability issues. Whenever either the size of the network, or the requested demand of the study at hand increases, current computational requirements (memory and CPU speed) just not suffice anymore. To overcome this, either the heuristics or the column generation technique can be used. The heuristics provide a solution in a reasonable time frame, in our assumed case studies with a 5% optimality penalty. The column generation technique is able to decrease the optimality gap to 2%, providing solutions within acceptable time limits. This is achieved by decomposing the problem at hand in two separate problems which are solved iteratively:

1. **The master problem.** Given a set of configurations (primary and backup paths), find the optimal combination to satisfy all demands.

2. **The pricing problem.** An identification of an improving configuration, which decreases the current value of the objective of the master problem.

Which solution technique to use depends on (i) solution optimality, (ii) the number of requested connections and (iii) computation time frame.

If optimality is of primary concern for (very) low load conditions, the ILP technique could be advised. However, if computation time can be exchanged for optimality, the heuristic is advised for average load conditions while, under the same computation/optimality characteristics, the column generation technique is to be used for high load conditions.

Lastly we mention that no conclusive outcome could be formulated on the relation between column generation execution time and the number of IT end points in the network. On the contrary, the degree of the topology is a lot more influential, where a highly meshed network severely penalizes the execution time for the column-generation technique since the number of possible paths increases.

7.1.3 Energy considerations in optical grids/clouds

Chapter 5 focused on the energy expenditure of an integrated network and IT infrastructure, which is typical for cloud and grid architectures. Distributed computing is already seen as an energy-efficient architecture.

- End users only need low-power devices, since processing power (and hence also a large part of energy consumption) is moved into the network. This thin client setup has a significantly lower power consumption compared to e.g., desktop PC architectures [1].

- Distributed architectures provide aggregation points for workloads that would otherwise be run on separated devices, which means that via statistical multiplexing demands can be consolidated and hence hosts can be better exploited.

On top of these benefits, we aimed to optimize the coordinated allocation of optical network and IT resources to reduce overall energy consumption. Although past works have performed energy-minimizing studies on the different components of optical grids/clouds (focusing either on network or IT resources), the integration of them into one formalism had not yet been investigated.

We started with formulating a detailed, integrated energy model for both the network and IT resources. This energy model was then used in our proposition for an online heuristic that for a given request finds (i) an IT end point able to process the request (the scheduling problem) and (ii) a route from the requesting source to that IT end point in the optical network (the routing problem). Again, we exploit the anycast principle, while allowing unused resources to be put into a sleep mode, not consuming any energy.

When trying to minimize either pure IT or network energy, we noted that in low load conditions, it is better to try to switch off data centers, as they consume a huge amount of energy. However, as data centers need to be started to accommodate a higher number of requests, the large focus on IT energy minimization leads to a suboptimal energy use of the network: from then on intelligently routing leads to a significant reduction of total energy of about 3% compared to IT-only optimization.

A larger reduction in energy consumption can be achieved by considering the energy parameters of both network and IT resources jointly. A careful consideration of these values, leads to even better resource allocations (and respective energy consumption values) than either the pure IT or pure network energy optimization. Moreover, we indicated that for topologies with a reasonable network degree, this energy reduction does not necessarily lead to a service blocking penalty. Lastly, we have shown that our unified algorithm, which computes the destination and route to that destination in one step, outperforms present-day algorithms considering IT resources first (planning) and subsequently the network (routing), in particular for low to average load conditions.

7.1.4 A scalable control plane for optical grids/clouds

The control mechanism of an integrated network and IT infrastructure to select both the data center (IT end point) and the network resources (the path to that IT end point) becomes critical for guaranteeing that infrastructure's efficient operation. Hence, Chapter 6 introduced a set of extensions to a Generalized Multi-Protocol Label Switching (GMPLS) and Path Computation Element (PCE)-based network control plane, referred to as NCP+, to enable anycast path computation for NIPS requests in a multi-domain optical scenario. We proposed (i) new network control plane modules to disseminate and process the necessary IT resource information in the NCP, (ii) main extensions to existing GMPLS and PCE pro-

tocols and (iii) path computation and topology representation algorithms for the PCEs and evaluate them in terms of computation time, average resource load and service blocking in simulation case studies.

Our NCP+ proposal adopts a hierarchical architecture where a parent PCE is in charge of coordinating the end-to-end (e2e) path computation through multiple intra-domain requests to its child PCEs. In this model there is one centralized point (the Parent PCE+ server) which maintains a global view (without any internal details) of all the domains, including the attached IT resources. This view is called the aggregated view of the topology. We discussed and compared two proposals for aggregation schemes: Full Mesh (*FM*) and Star aggregation (*Star*). The former methodology represents the domain as a complete graph between the border nodes and all IT sites. The latter introduces a virtual node, aggregating the capacities of the IT sites, which is bi-connected to all border domain nodes. The network information is aggregated and stored for each link leaving a border node. The advantage of *FM* is that it is able to compute paths with a more detailed view on the topology than *Star*, at a computation time penalty. *Star* however, is faster with respect to computing time, but generates sub-optimal paths as it stores less detailed information of the complete infrastructure.

We showed that the *FM* aggregation scheme has the best performance regarding service blocking, even though it comes at a cost in complexity (path computation time).

1. It reduces the service blocking.
2. It computes shorter hop paths.
3. Although the path computation time for *FM* is larger than for *Star*, this advantage of *Star* over *FM* is nullified as the total setup time is proportional with the path length, which on average is greater for *Star* than for *FM*.

7.2 Future directions

Results and conclusions from this research work suggest some possible future work.

- **Application of physical layer impairments.** In the context of energy efficiency in optical grids/clouds, the issue of physical layer impairments and their effects on the quality of transmission needs to be investigated. The authors of [2] have already demonstrated that focussing on the signal quality without energy considerations increases the energy consumption considerably, while exclusively optimizing the energy consumption degrades the signal quality significantly. Their proposal for a combined approach (optimizing energy and the signal quality at the same time) shows that they are able

to significantly reduce power consumption while guaranteeing the required signal quality. We believe that it is possible to reach even better results in a cloud context, where we consider IT and network resources jointly, as there is more freedom in choosing the IT end points and the network resources towards them.

- **Online resilient scenario.** The solution techniques proposed in Chapter 4 all assume a fixed input (the source vector). However, in a dynamic scenario, requests arrive sequentially and consequently, need to be provisioned on demand. We need to port the solution techniques from an offline to an online scenario and investigate their computation time and corresponding blocking ratio. We believe that the heuristics can be efficiently used in the online scenario as they compute the solution for the batch of requests sequentially (i.e., each request is added one at a time) and similar results can be expected.

- **Combination of resiliency and EE.** In our work, we have treated the issues of resiliency and energy efficiency for an integrated network and IT infrastructure separately. However, sharing of unused backup capacity also allows it to be put into an inactive state, not consuming any energy. The authors of [3] have already demonstrated that deactivating the backup network resources, when using dedicated path protection, allows for a network energy reduction of 34% (compared to leaving them in an active mode). We need to extend this idea to a cloud context in three ways:

  - Consider both the energy consumption of network and IT resources.
  - Investigate shared path protection, as this case employs fewer resources.
  - Allow also sharing of IT backup capacity, i.e., a backup virtual machine/computing node can be shared as long as the primary IT and network resources for both requests are not the same.

  We need to investigate the advantages (possible energy reduction and resource savings) while also looking into the effect on the service (as inactive resources require a boot time to start).

- **Resiliency and EE in a multi-domain scenario.** Our work on resiliency and energy efficiency has provided solutions for a scenario where the infrastructure is managed by one entity (single domain). We need to adapt those strategies in order to apply them in our multi-domain NCP+ proposition. Challenges include:

  - The selection of parameters which need to be aggregated and sent to the parent PCE (using PCEP) to allow for EE provisioning. One direction could be to add an extra parameter, expressing the energy required to reach a certain node (data center or border node). Based on

the availability and the associated energy consumption estimation, the PCE Parent could provision energy efficiently.

  - Investigate the influence of computing link distjoint paths in the aggregated topology of the PCE parent and the child domains. This is a complex problem for which predictions are difficult to make.

- **Follow the sun, follow the wind.** In the context of EE, moving workloads from one data center to one which has access to instantaneous renewable energy, needs to be investigated (see for instance [4]). The combination of putting unused resources into a sleep mode, while active resources are constantly relocated to these green locations, would require an extremely dynamic and flexible management system.

- **Large Scale Emulation studies.** Our work has been evaluated by means of simulation. However, to excite and convince the business community of the value of our work, large scale emulations will be necessary. In emulations, actual implementations are tested in a controlled environment, indicating the actual and real benefits of the proposed solutions.

References

[1] W. Vereecken, L. Deboosere, P. Simoens, B. Vermeulen, D. Colle, C. Develder, M. Pickavet, B. Dhoedt, and P. Demeester. *Power efficiency of thin clients*. In Proc. Fut. Netw., 3rd Ghent University-KEIO joint workshop, pages 1–1, Ghent, Belgium, 10 Apr. 2010.

[2] C. Cavdar, M. Ruiz, V. L. Monti, Paolo, and L. Wosinska. *Design of green optical networks with signal quality guarantee*. In IEEE Int. Conf. on Commun. (ICC), pages 1–6, Ottawa, Canada, 10–15 Jun. 2012.

[3] A. Jirattigalachote, C. Cavdar, P. Monti, L. Wosinska, and A. Tzanakaki. *Dynamic provisioning strategies for energy efficient WDM networks with dedicated path protection*. Optical Switching and Netw., 8(3):201–213, Mar. 2011.

[4] K. Nguyen, M. Cheriet, M. Lemay, B. Arnaud, V. Reijs, A. Mackarel, P. Minoves, A. Pastrama, and W. Heddeghem. *Renewable energy provisioning for ICT services in a future internet*. In J. Domingue, A. Galis, A. Gavras, T. Zahariadis, D. Lambert, F. Cleary, P. Daras, S. Krco, H. Mller, M.-S. Li, H. Schaffers, V. Lotz, F. Alvarez, B. Stiller, S. Karnouskos, S. Avessta, and M. Nilsson, editors, The Future Internet, volume 6656 of *Lecture Notes in Computer Science*, pages 419–429. Springer Berlin Heidelberg, 2011.

A

Providing resiliency for optical grids by exploiting relocation: a dimensioning study based on ILP

Buysse, J.; De Leenheer, M.; Dhoedt, B. & Develder, C., *Providing resiliency for optical grids by exploiting relocation: A dimensioning study based on ILP*, published in Computer Commununications, Vol. 34, pp. 1389-1398, 2011

This paper has been included in the dissertation, as it forms the base for the work described in Chapter 4.

Abstract Grids use a form of distributed computing to tackle complex computational and data processing problems scientists are presented with today. When designing an (optical) network supporting grids, it is essential that it can overcome single network failures, for which several protection schemes have been devised in the past. In this work, we extend the existing shared path protection scheme by incorporating the anycast principle typical of grids: a user typically does not care on what specific server this job gets executed and is merely interested in its timely delivery of results. Therefore, in contrast with classical shared path protection (CSP), we will not necessarily provide a backup path between the source and the original destination. Instead, we allow to relocate the job to another server location if we can thus provide a backup path which comprises less wavelengths than the one CSP would suggest. We assess the bandwidth savings enabled by relocation in

a quantitative dimensioning case study on an European- and an American network topology, exhibiting substantial savings of the number of required wavelengths (in the order of 20%–50%, depending on network topology and server locations). We also investigate how relocation affects the computational load on the execution servers. The case study is based on solving a grid network dimensioning problem: we present integer linear programming (ILP) formulations for both the traditional CSP and the new resilience scheme exploiting relocation (SPR). We also outline a strategy to deal with the anycast principle: assuming we are given just the origins and intensity of job arrivals, we derive a static (source, destination)-based demand matrix. The latter is then used as input to solve the network dimensioning ILP for an optical circuit-switched WDM network.

A.1 Introduction

A.1.1 Optical grids

The very demanding network and IT requirements of several problems in domains ranging from astrophysics [1], climate modeling [2] and fluid dynamics [3] have led to the conception of grid computing. A grid consists of different heterogeneous resources (computational, storage and networking) which are geographically spread over various administrative domains, implying that resource coordination is not subject to centralized control. To interconnect the distributed resources, Optical networks with Wavelengths Division Multiplexing (WDM) are a suitable candidate for it, since they can support high bandwidth traffic with low latency in a reliable way. This has led to the concept of optical grids or so-called lambda grids [4, 5]. While multiple alternative optical switching techniques have been proposed (including optical burst switching, OBS), in this paper we focus on circuit-switched (OCS) optical grids where wavelengths connections (so-called lambdas in lambda-grids) are set-up, establishing connectivity between a source- and a destination-node using a two-way reservation.

One characteristic of an Optical grid is the *anycast principle* which in this context means that the user is not interested in the location of the execution of his application (which we will denote as jobs), but is merely concerned with the successful execution of the jobs subject to predetermined requirements such as a fixed deadline or some other quality guarantee. To guarantee this timely delivery, we have to make sure that it is also realized in case of a resource failure (either network- or computing resources). In this work we address survivability of single link failures in the optical network. There are two basic strategies to protect an optical network, namely *restoration* and *protection* [6]. The former is a reactive procedure where connections affected by a failure are routed along an alternative path that is calculated and set up at the time of the failure. In case of protection,

the backup path is pre-computed. This paper discusses two protection schemes, establishing for each primary path has an associated backup path to be used whenever one of the links in the primary fails. The first protection scheme we take into consideration is the well-known scheme we denote as Classical Shared Path (CSP) protection: wavelengths can be shared among backup paths, as long as the corresponding primary paths are link disjoint. (Its counterpart, *dedicated Path* protection, does not allow this sharing.) Our proposed second scheme, Shared Path protection with Relocation (SPR) is an extension of the CSP scheme, where instead of reserving a backup path to the end point of the primary path — being the original destination as determined by the grid scheduler — we can provide a backup path to another (possibly closer) server site, hence allowing the jobs to relocate. We quantitatively assess the benefits in terms of overall number of wavelengths used on the whole of all network links (i.e. achievable network load reduction, NLR), as well as the potential penalty in terms of extra load on the servers receiving the relocated jobs.

To achieve these results, we show how to solve the network dimensioning problem by means of an Integer Linear Program (ILP). ILPs are presented for both Classical Shared Path protection (CSP) providing a backup path to the original end point, and Shared Path protection with Relocation (SPR). Traditionally, a static demand matrix serves as input for these formulations, specifying the number of connections to set-up between each source and possible destination. However, in a grid scenario, the destination of jobs is left up to the grid scheduler (cf. anycast). Hence, we will consider a dimensioning approach starting from arrival rates specifying the job intensity per source. In Section A.3 we outline a phased strategy to convert these arrival rates to a static (source, destination)-based demand matrix. Thereby, we use an ILP to find the best possible locations for the server sites. After this, we analytically compute the server capacity while meeting a predefined job loss rate. As a last step, we use simulation, assuming a certain scheduling policy, to find the resulting static demand matrix specifying the job rates exchanged between each (source, destination)-pair.

The remainder of this paper is structured as follows. First, in Section A.2, we briefly discuss the possible failures which can occur in optical grids. In Section A.3 we explain how to obtain a (source,destination)-based traffic matrix from a grid scenario only specifying job origins. In Section A.4.1 we present Integer Linear Programming (ILP) formulations for dimensioning the network assuming the new SPR protection scheme, as well as the CSP benchmark case. We present an evaluation of these models by a case study in Section A.5. Final conclusions are summarized in Section A.6.

A.1.2 Related work

In [7] a survey is presented based on input of the grid community sharing their actual experience regarding fault treatment. It shows that a large part of the failures originate from hardware deficiencies ($\pm 35\%$), indicating the importance of our study. The relevance of the considered single link failure model is demonstrated in [8]. The authors state that in order to provide complete protection from all dual-link-failures, one may need almost thrice the spare capacity compared to a system that protects against all single-link failures. However, it has also been shown that systems designed for 100% single-link failure protection can provide reasonable protection from dual-link failures.

A large research effort has been devoted to recovery strategies resolving resource (i.e. grid server) failures. There are two strategies which aim to improve the system's performance in the presence of failure: job checkpointing and replication. Job checkpointing [9, 10] periodically stores an image of the running job, which can be restored in case of a failure. In replication [11, 12] a job is sent to a primary server and to a set of replication servers. In case of a failure of the primary server, its role is taken over by a replication server which continues the job execution.

In [13] several adaptive heuristics, based on both approaches and their combination were designed and evaluated. The results have shown that the overhead of periodic checkpointing can significantly be reduced when the checkpointing frequency is dynamically adapted as a function of resource stability and remaining job execution time. Furthermore, adaptive replication-based solutions can provide for even lower cost fault-tolerance in systems with low and variable load, by postponing replication according to system parameters. Finally, the advantages of both techniques are combined in the hybrid approach that can best be applied when the distributed system properties are not known in advance. Note that [13] disregards network failures, and uses a simplified network model.

In this paper, we will focus on the network aspects and consider protection against network failures (and as such is complementary to server resiliency strategies as checkpointing and replication). For a review and classification of the main optical protection techniques for the WDM-layer, we refer to [14]. We will evaluate our proposed relocation strategy SPR by formulating two ILPs. ILPs have been widely exploited in previous works to find a optimal solution to a certain network design and planning problem. The main advantages of these kind of formulations is the easy way of adapting the description of the network environment: cost functions, wavelength conversion, protection scheme etc.

These ILP formulations can be divided into two categories: Flow Formulation (FF) and Route Formulation (RF). The authors of [15] have investigated these formulations in unprotected networks to conclude that although they have the same computational complexity, RF has the advantage of reducing the number of vari-

ables by imposing a restriction on the number of allowable paths between a source and a destination. In [16] the authors focus on the computational efficiency of the ILP model in order to provide a more effective tool for planning. The formulation exploits flow aggregation and consists in a new ILP formulation that can reach optimal solutions with less computational effort compared to other ILP approaches. Yet, the solution of the so-called source formulation ILP in [16] requires a post-processing step to find the actual routing and wavelength assignment (RWA) and it does not consider resilient network dimensioning.

In this paper (which is an extended version of [17]), we stick to the traditional source-destination method based on flow formulation, where the CSP case is largely based on the ILP presented in [18]. There the authors investigate the problem of fault management in a meshed WDM network with failures due to fiber cuts: both ILP and heuristic solutions are examined and their performance is compared through numerical examples.

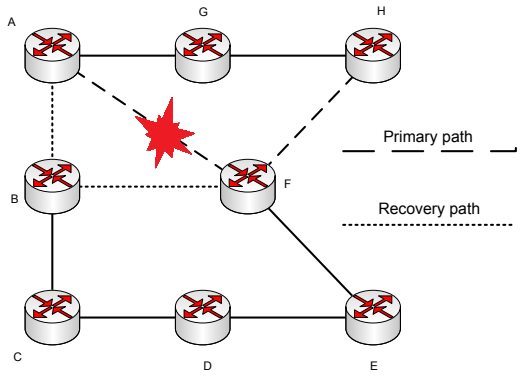
A.2 Failures in optical grids

Network failures in optical networks are either known in advance (planned failures) and some preventive measures can be taken to overcome them, or they are unplanned and caused by erratic events such as natural disasters, fiber cuts, etc. From a network provider's point of view, it is impossible to devise pre-planned protection schemes for all imaginable network failures, and hence the most occurring failures are split up into various restricted failure scenarios to be overcome in a gracious manner. For network resources, typically cable cuts and equipment failures are the most frequent and two scenarios are considered:

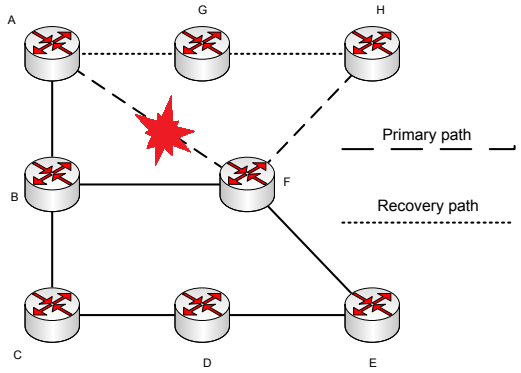
1. *Single link failure*: a link between two adjacent network nodes fails and consequently no information can be sent between them. Schemes protecting against these kind of failures can reroute around the end nodes of the failed link (Fig. Fig. A.2a) or find a new path from the source to destination (Fig. A.2b).

2. *Single node failure*: a network element fails and hence all its incident links are out of service (Fig. A.2c).

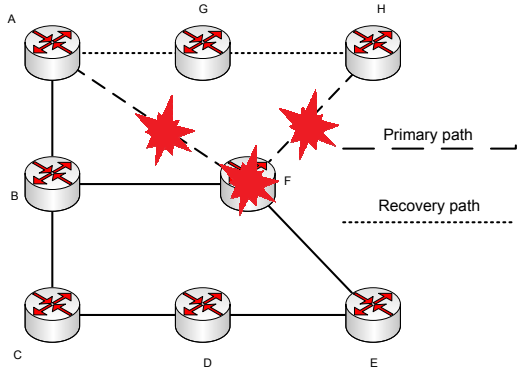
The aforementioned protection scheme, Classical Shared Path protection (CSP), is subdivided in the first failure class as is our newly proposed scheme, Shared Path protection with Relocation (SPR), for the very reason that it is an extension of CSP. We denote a primary path as the path which is used in the failure free scenario and it corresponding backup path as the path which is used when a single-link failure occurs on that primary path. As indicated before, we are dealing with a Shared path protection scheme which indicates that two primary paths P_1 and P_2 can be



(a) Single link failure with link protection: When the link A-F fails, this link is bypassed by the links A-B and B-F after which the original path is reused.



(b) Single link failure with path protection: When a link on the path from A to H fails, the backup path A-G-H is taken.



(c) Single node failure, causing two links to fail. When node F fails, the recovery path A-G-H is taken.

Figure A.1: Failure scenarios and recovery paths in a communication network for a connection from A to H.

protected by two partially overlapping backup paths B_1 and B_2 as long as P_1 and P_2 are link disjoint (Eq. A.1).

$$R_1 \cap R_2 \neq \emptyset \Rightarrow P_1 \cap P_2 = \emptyset \tag{A.1}$$

A.2.1 Shared path protection with relocation

In the CSP scheme, a primary path and its corresponding backup path end at the same node (in this case some grid server site) and two backup paths can share wavelengths as long as their corresponding primary paths are link disjoint. We will relax the first constraint so that the endpoints of a primary- and backup path can end in different grid server sites, as to potentially reduce the network load. This implies relocation of grid jobs from the primary server site to an alternate site for which we could create a backup path comprising fewer hops (not including any of the primary links) or finding a backup path where more wavelengths can be shared (i.e. inferring no additional cost because they are already installed for another backup path). This relocation is possible by the grid specific anycast principle: when a user creates a job, several resources are able to execute it and only one of them is chosen, generally by the grid scheduler. Hence, as illustrated in Fig. A.2, in case of a network failure on the primary path we could relocate the job to another computing resource. Still, this could cause a trade-off between lowering network resources (fewer wavelengths) and potentially increasing resource capacity: we have to cater for extra computing power at the relocation server to process relocated jobs. Note however that such additional server capacity will be required anyhow to cope with grid resource failures.

A.3 Deriving a (source,destination) traffic matrix from anycast grid traffic

Our goal is to evaluate the above-mentioned relocation scheme against Classical Shared Path protection, from a network dimensioning perspective. Hence, we will employ ILP formulations to derive the required amount of wavelengths needed to equip for a given connection demand between (source,destination)-pairs. However, in an optical grid scenario where the anycast principle applies, the traffic is rather specified by the number of jobs arriving at given source sites and the destination can essentially be freely chosen among server sites. Hence, we need to convert this anycast traffic specification to a clearly defined (source, destination)-based traffic matrix as required for network dimensioning algorithms (such as ILP). We now will present a methodology realizing this conversion, before discussing the network dimensioning in Section A.4.

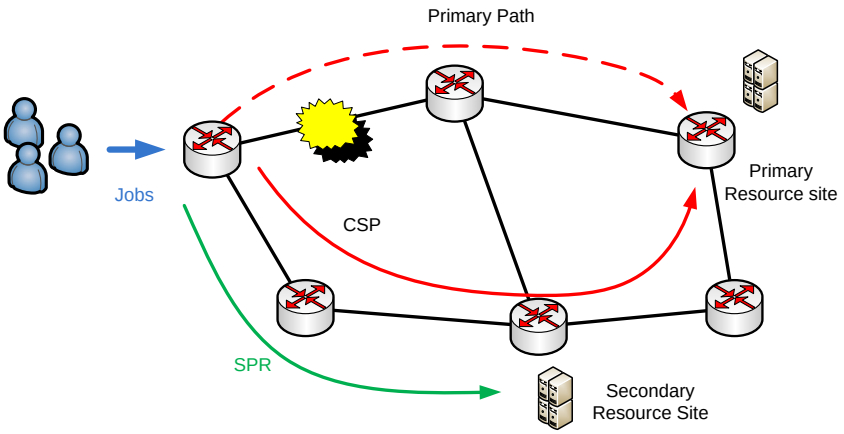


Figure A.2: In a Classical Shared Path protection scheme (CSP) a primary path is protected by a link disjoint backup path. By allowing the backup path to end in a server different from the primary server, we can achieve a network load reduction. This resilience scheme is called Shared Path protection with Relocation (SPR).

To obtain our traffic matrix, we resorted to an iterative approach. This is discussed in detail in [19], and summarized below. The subsequent phases followed stem from the realization that three aspects are important when trying to obtain a (source, destination)-based traffic matrix from the demand vector:

1. The location of the grid server sites, which are capable of executing the jobs.
2. The amount of servers at each of the chosen server sites.
3. The scheduling algorithm: the policy the grid management enforces to distribute the jobs among the different server sites.

Thus, the first steps are to decide where to locate the server sites and how many server CPU's to install at each site (e.g. while meeting a maximum job loss rate criterion).

A.3.1 Find the K best server locations

Choosing the optimal choice for the server locations is a K-medoid problem: the goal is to find K clusters, where the nodes in each cluster are grouped together according to a specified metric and where the cluster centers represent the chosen server sites. We have formulated this as a compact ILP shown below, making the simplifying assumption that site i sends all its jobs to the same server (which may not be the case in reality, depending on the scheduling policy, described in Section A.3.3).

The decision variables deciding on the server site locations are:

- $T_j = 1$ if and only if site j is chosen as a server site location, else 0.
- $S_{i,j} = 1$ if and only if site j is the target server for traffic from site i, else 0.

The given input parameters to base these decisions on are:

- λ_i is the job arrival rate at site j $(i = 1 \ldots N)$.
- $H_{i,j}$ is the routing distance (typically hop count) from site i to site j $(i, j = 1 \ldots N)$.
- K is the number of server sites to choose.

The objective function of the ILP is given in eq. (A.2), the constraints are in eq. (A.3)-(A.5).

$$min\left(\sum_i \sum_j \lambda_i \cdot H_{i,j} \cdot S_{i,j}\right) \tag{A.2}$$

$$\sum_j T_j = K \tag{A.3}$$

$$\sum_j S_{i,j} = 1 \quad \forall i \tag{A.4}$$

$$S_{i,j} \leq T_j \quad \forall i, j \tag{A.5}$$

A.3.2 Determining the server capacities

We continue with dimensioning the processing power at each server site, i.e. the number of CPUs. We have made some assumptions which appear to be realistic [20]: we assume Poisson arrivals and exponentially distributed service times. With these assumptions we solve the well-known ErlangB formula Eq. A.6 to establish the total number n of servers needed to meet a maximum job loss rate of $x\%$. We subsequently distribute that amount of n CPUs among the server sites, proportionally to the cluster arrival rate at each server site (thus installing the most CPUs where the most traffic is arriving, as [19] indeed showed this choice results in lower network loads).

$$ErlangB(\lambda, \mu, n) = \frac{\frac{\left(\frac{\lambda}{\mu}\right)^n}{n!}}{\sum_{k=0}^{n} \frac{\left(\frac{\lambda}{\mu}\right)^k}{k!}} = x \tag{A.6}$$

A.3.3 Scheduling policy

We have adopted a *mostfree* scheduling policy (see [19]): first try the server nearest (in terms of hop count, hence denoted as 'local' server site) to the job's originating site. If this 'local' server site is not available, then choose a free CPU at server site f, where f is the server site with the highest number of free server CPUs, in an attempt to avoid overloading sites and thus limiting non-local job execution. In this step we have resorted to simulations because of the anycast principle: it is hard to obtain accurate estimates for the inter-site traffic using analytical techniques (although that under certain assumptions, numerical calculation can be achieved [21]). Note that this scheduling policy holds at runtime and so the assumption that each source site sends to the same server made in Section A.3.1 does not necessarily hold. Yet, if the number of servers is appropriately chosen, the majority of the jobs should end up being executed at the closest server (see [19]).

After this step we know how many jobs are exchanged between every grid node pair in the considered network. By appropriately scaling with the job data sizes and rounding these numbers, we finally end up with a demand matrix containing a number of connections between each grid node pair.

A.4 Network dimensioning model

We investigate a network design model with a static traffic matrix in which a known set of connection requests is assigned to the network. Each connection represents a point-to-point light path (circuit) from a source to a destination, able to transport a given capacity. Furthermore, we assume in this paper a so-called virtual wavelength path (VWP) network [18], implying that all optical cross-connects (OXC) are able to perform wavelength conversion. Note that if OXCs do not support wavelength conversion, the wavelength continuity constraint must hold and the resulting network is a plain wavelength path (WP) network.

Our topology is modeled as a graph $G = (V, E)$ where the links are represented by a directed edge $(i, j) \in E$ (with $|E| = L$), while the vertices $v \in V$ (with $|V| = N$) represent the OXCs. The static traffic matrix is converted into a list of connection objects $\beta = \{\phi_1, \phi_2, \ldots \phi_n\}$ where a connection ϕ_c corresponds a unit demand requiring a single wavelength path, identified by its index c. Two connections can have the same source and the same destination.

We define the following variables:

- $p^{\phi}_{i,j}$: binary decision variable which is 1 if link (i, j) is used for the primary path for connection ϕ.
- $r^{\phi}_{(i,j)}$: binary decision variable which is 1 if link (i, j) is used as part of a protection path for connection ϕ

- m_j^{ϕ}: binary decision variable which is 1 if node j is a backup resource which is used for connection ϕ.
- $\pi_{i,j}$: integer auxiliary variable, the total number of wavelengths on link (i,j) used for a backup path.
- $P_{i,j}$: integer auxiliary variable, the total number of wavelengths on link (i,j) used for a primary path.
- $\Theta^{\phi}_{(i,j),(k,l)}$ is an integer variable introduced to calculate the number of shared wavelengths.

A.4.1 ILP formulation

The objective function Eq. A.7 expresses that we want to minimize the the total number of primary and backup wavelengths:

$$\min \left(\sum_{i,j} \pi_{i,j} + \sum_{i,j} P_{i,j} \right) \tag{A.7}$$

Constraints Eq. A.8 express the demand constraints and flow conservations for the primary paths. When j is the source node of connection ϕ ($j = s$) then we should only have a flow originating from that source. If j is the destination of ϕ ($j = d$) then this node should be the ending node of the flow. In the last case, where the j is an OXC, any connection arriving should also leave again. Similarly, the constraints (A.9) are the flow conservations for the backup paths, where the m_j^{ϕ} variable will decide which node is the destination and will depend on whether we are considering CSP or SPR.

$$\sum_{i:(i,j)\in E} p^{\phi}_{(i,j)} - \sum_{k:(j,k)\in E} p^{\phi}_{(j,k)} = \begin{cases} -1: & j = s \\ +1: & j = d \\ 0: & \text{else} \end{cases} \qquad \forall \phi \in \beta, \forall j \in V \tag{A.8}$$

$$\sum_{i:(i,j)\in E} r^{\phi}_{(i,j)} - \sum_{p:(j,p)\in E} r^{\phi}_{(j,p)} = \begin{cases} -1: & j = s \\ m_j^{\phi}: & \text{else} \end{cases} \qquad \forall \phi \in \beta, \forall j \in V \tag{A.9}$$

The next constraints (A.10) express that a primary path and a backup path cannot overlap.

$$r^{\phi}_{(i,j)} + p^{\phi}_{(i,j)} \leq 1 \qquad \forall \phi \in \beta, \forall (i,j) \in E \tag{A.10}$$

In Eq. A.11 we introduce the binary variable $\Theta^{\phi}_{(i,j),(k,l)}$ which is 1 if and only if for connection ϕ link (k,l) is protected by link (i,j). These variables are used in Eq. A.12 to bound the $\pi_{(i,j)}$ variables which count the shared backup wavelengths for a link (i,j).

$$\begin{aligned} \Theta^{\phi}_{(i,j),(k,l)} + 1 \geq r^{\phi}_{(i,j)} + p^{\phi}_{(k,l)} \\ \forall \phi \in \beta, \forall (i,j),(k,l) \in E \end{aligned} \tag{A.11}$$

$$\begin{aligned} \pi_{(i,j)} \geq \sum_{\phi} \Theta^{\phi}_{(i,j)(k,l)} \\ \forall\,(k,l) \in E, \forall\,(i,j) \neq (k,l) \in E \end{aligned} \tag{A.12}$$

In the case of CSP we enforce that the primary server and backup server need to be the same by eq. Eq. A.13.

$$m^{\phi}_{j} = \begin{cases} 1 & \text{if } j \text{ is the primary server of } \phi \\ 0 & \text{else} \end{cases} \tag{A.13}$$

On the other hand, to achieve SPR we replace Eq. A.13 with Eq. A.14-Eq. A.15 to let the ILP freely decide which backup server to use.

$$\sum_{\delta \in \Delta} m^{\phi}_{\delta} = 1, \quad \forall \phi \in \beta \tag{A.14}$$

$$m^{\phi}_{\delta} = 0, \quad \forall \delta \notin \Delta \tag{A.15}$$

A.4.2 Complexity

According to [18], the complexity of an ILP heavily depends on the number of variables and to a lesser extent on the number of constraints. The number of variables for in the ILP formulations are the same for both the CSP and SPR cases, while only the number of constraints differ. Nevertheless, there is a big difference in the running time of the SPR vs. the SPR: running a CSP instance with the same input parameters takes much longer than an instance of SPR.

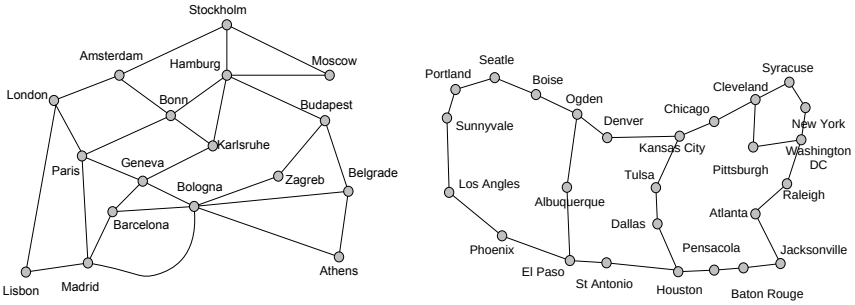
The number of variables is

$$2\,|E| \times (|\beta| + 1) + |\beta| \times (|V| + |E|^2)$$

and depends mostly on the number of desired connections and the topology. The number of constrains for CSP is

$$|\beta| \times (2\,|V| + |E|) + |E|^2 \times (|\beta| + 1)$$

If we want to achieve SPR we have add $|\beta| + |\Delta|$ more constraints. We notice that the ILP is not very scalable (quadratic in the number of links) and will not suffice to deal with larger instances.



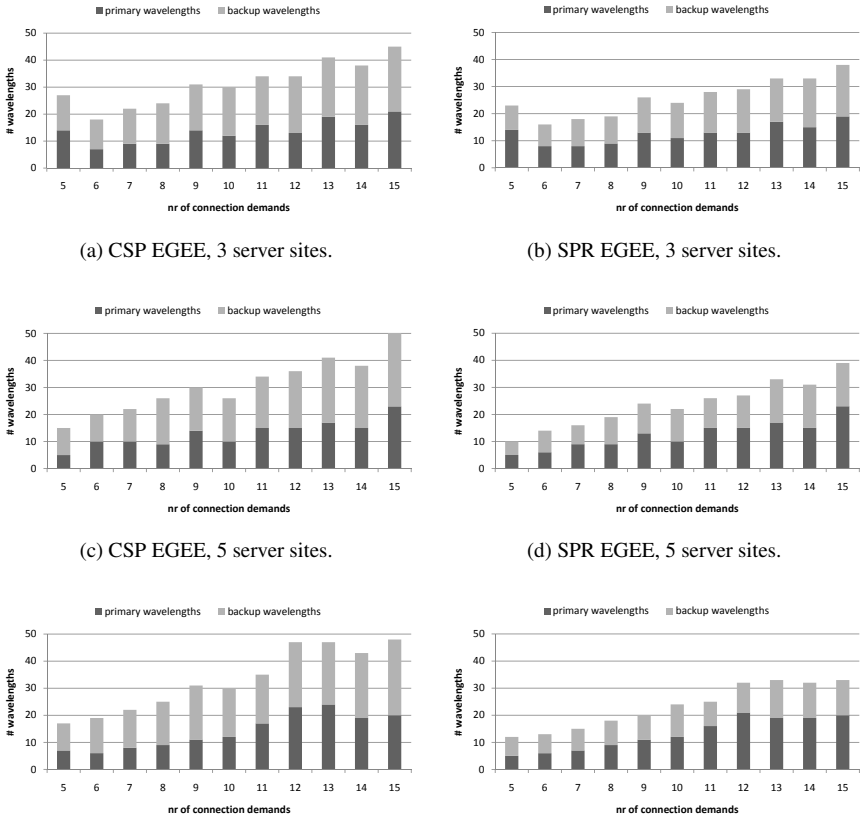
(a) EGEE topology

(b) USNLR topology

Figure A.3: Topologies for the case studies. The first is based on the EGEE GEANT network consisting of 17 nodes and 54 links. The second is the US National Lambda Rail (USNLR) consisting of 27 nodes and 60 links.

A.5 Case study

We have considered the two topologies depicted in Fig. A.3, where each link is supposed to be bidirectional. Fig. A.4a is based on the Géant 2 network topology and its associated various national research- and education networks (NRENs) and consists of 17 nodes and 54 links. Fig. A.4b is based on the National Lambda Rail (NLR) which provides a testbed for advanced research at over 280 universities, U.S. government laboratories and advanced programs across the United States and consists of 27 nodes and 60 links. For each topology, We have generated 10 random arrival rate files, containing for every possible source site the rate of jobs it needs to send out. By applying the strategy explained in Section A.3 we end up with 10 different demand matrices (with increasing number of connections) for each network with respectively 3, 5 and 7 server sites. These static demand matrices served as input for the ILP and their results are presented in the sections below. (Note that for a given number of unit connection demands we chose not to present average results over multiple random instances, since the chosen server sites may differ among them.) We will use the notation N_x^y as a network with x server sites and a connection demand of y connections.



(a) CSP EGEE, 3 server sites.

(b) SPR EGEE, 3 server sites.

(c) CSP EGEE, 5 server sites.

(d) SPR EGEE, 5 server sites.

(e) CSP EGEE, 7 server sites.

(f) SPR EGEE, 7 server sites.

Figure A.4: The total number of wavelengths for both the CSP and SPR case, for the EGEE network with 3, 5 or 7 server sites. Although there is little or no difference in the amount of primary wavelengths between both CSP and SPR, the number of backup wavelengths for SPR amounts to only around 50% of the number of backup wavelengths of CSP. Note that each bar is the result of a single dimensioning outcome, hence the non-monotonic increase for increasing number of unit connection demands.

A.5.1 Influence of relocation

We first discuss the results obtained for the EGEE Géant-based network. In Fig. A.4 we plot the total number of wavelengths, summed over all links, being used for all the primary- and backup paths. As expected, with an increasing load, the required network resources tends to grow. (Note that the increase is not monotonic, given that we are considering single random cases: thus it is possible that comparing two cases with different number of unit connection demands, the one with the higher demand not necessarily requires more wavelengths.)

Comparing the amount of primary wavelengths used in CSP with the amount of primary wavelengths in SPR we see that there is little or no difference and this observation is independent on the number of server sites which have been chosen. This means it does not often happen that SPR finds a primary path (different from the CSP case) to create more opportunities for sharing wavelengths among different connections' backup paths.

Yet, the number of backup wavelengths can be drastically decreased by employing relocation (SPR requiring on average in the order of 50% fewer backup wavelengths than CSP). There are two possible reasons (which may apply simultaneously) why relocating to another site consumes fewer backup wavelengths:

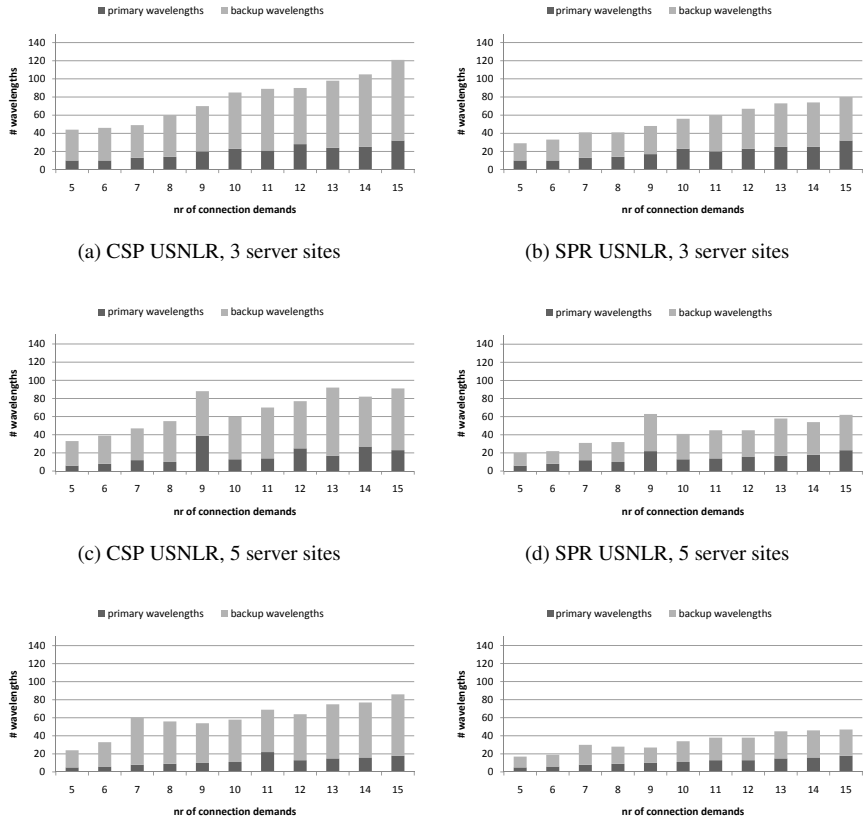
1. *Closer backup site:* Relocating a job allows to establish a backup path to a another (backup) server site which — considering a failure of any of the primary path's links — is closer in terms of hop count (and thus a fewer wavelengths summed over all links), e.g. a server that lies on CSP's backup path to the primary server.

2. *More sharing:* A connection ϕ's path to a server site, other than the primary one, could contain many backup wavelengths for connections having a primary paths disjoint from ϕ's. Hence, a larger portion of such a backup path may comprise wavelengths shared with others, requiring fewer wavelengths to be set-up exclusively for ϕ.

Looking at Fig. A.5 for the USNLR network, and comparing with the EGEE results, we observe substantial difference between the absolute numbers of wavelengths between the EGEE and the USNLR cases. This obviously stems from the highly different network topologies: the EGEE topology is more meshed while the USNLR topology is much sparser, resembling a composition of rings. Hence the cycle formed by a primary- and its corresponding backup path covers ring-like structures which comprise considerably more hops than in a highly meshed topology. Apart from the relatively higher number of backup wavelengths, similar observations as for the EGEE network can be made:

- With an increasing load, we generally achieve a higher number of required wavelengths.

- Comparing CSP with SPR, we see that we can drastically reduce the number of wavelengths.
  - This decrease is not induced by a decrease of primary wavelengths, because that number stays the same in most cases for CSP and SPR.
  - The decrease mainly stems from a reduction in backup wavelengths by either relocating to another closer server site or exploiting a sharing possibility which was not possible in the CSP case.



(a) CSP USNLR, 3 server sites

(b) SPR USNLR, 3 server sites

(c) CSP USNLR, 5 server sites

(d) SPR USNLR, 5 server sites

(e) CSP USNLR, 7 server sites

(f) SPR USNLR, 7 server sites

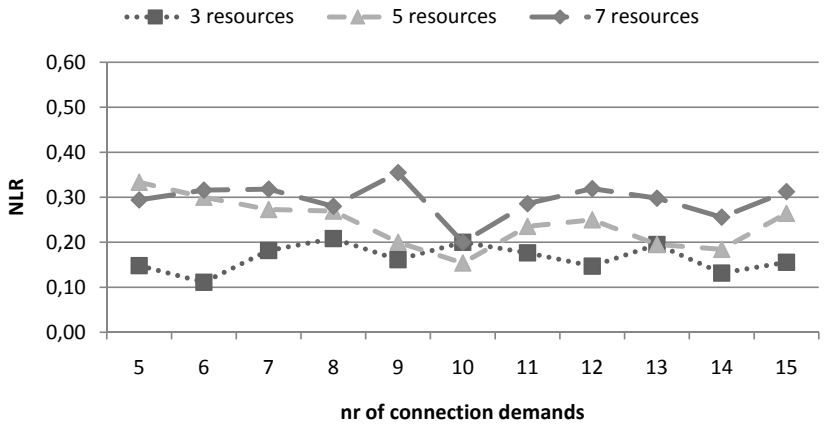
Figure A.5: The total number of wavelengths for both the CSP and SPR case, for the USNLR network with 3, 5 or 7 server sites. Similar observations apply as for the EGEE network.

A.5.2 Network load reduction

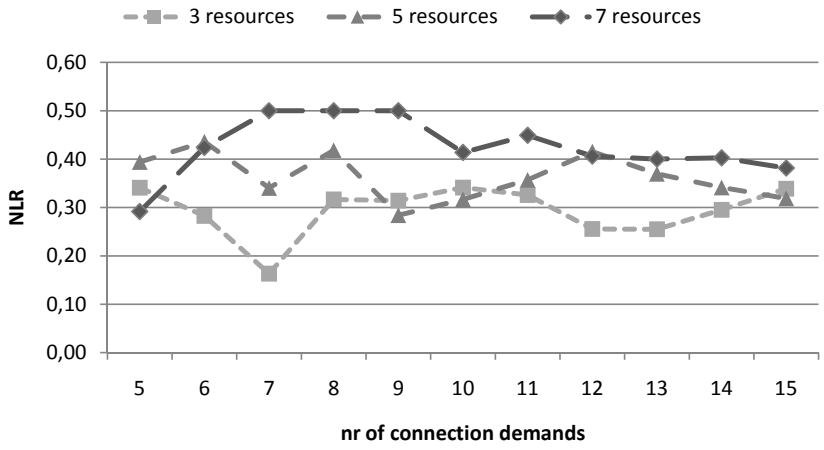
As pointed out in Section A.5.1, relocation achieves a lower number of consumed wavelengths — mainly induced by the decrease in backup rather than primary wavelengths — which we express formally as network load reduction (NLR) in Eq. A.16. We have plotted this NLR for both the EGEE- and USNLR network, in Fig. A.7a and Fig. A.7b respectively, for $N_x^y, x \in \{3, 5, 7\}, y \in [5, 15]$. We note that it seems that when employing more servers, the NLR increases. A reason for this may be that using more servers implies a higher probability of encountering another server on the backup path to the original one, and thus relocation is favorable. (Nevertheless, in some rare cases, having fewer servers does amount to a higher NLR; which may be due to single random demand creation, and the fact that having different server locations will amount to a different traffic matrix instance, cf. scaling to conform to integer demands.)

$$NLR = 1 - \frac{\text{total number of wavelengths SPR}}{\text{total number of wavelengths CSP}} \tag{A.16}$$

Considering the results for the USNLR network in Fig. A.7b, we note that qualitatively, the same observations apply as in the EGEE case. Yet, when comparing the NLR for a N_x^y case for both networks we notice that the USNLR more often than not has a larger NLR (up to 50%). The reason for this can be found in the topology structure. The EGEE is a more meshed topology (higher average node degree: 3.18 for EGEE vs. 2.22 for USNLR). Therefore in the USNLR case, the cycle composed by a primary path and its corresponding backup path will be quite long on average. Consequently, providing a backup path to another resource can drastically reduce the number of links necessary to that closer backup resource, especially when that new backup resources lies on the original backup path.



(a) EGEE topology



(b) USNLR topology

Figure A.6: The Network Load Reduction (NLR) achieved by relocation for both the topologies. By employing more servers sites we can achieve a higher NLR: for 7 server sites the savings achieved by relocation (SPR) compared to classical shared protection (CSP) are more substantial than for 5 or 3 server sites. Comparing both networks, we observe that in general we can achieve a higher NLR in the sparser USNLR topology.

A.5.3 Extra server capacity

As previously demonstrated, by relocating to another server site instead of the one originally (i.e. under failure free condition) proposed by the grid scheduler, a significant reduction in network resources can be achieved. But there is a trade-off: the relocation server receives more jobs than originally intended and thus, needs to reserve some spare capacity in order to execute the relocated jobs. Fig. A.7 shows for the EGEE topology the maximum amount of connections a server site receives for the cases with three (Fig. A.8a), five (Fig. A.8c) or seven server sites (Fig. A.8e) for the demand case of 15 unit connections. The black part is the load in failure free conditions, the grey part is the maximum of extra load it receives due to a single link fault.

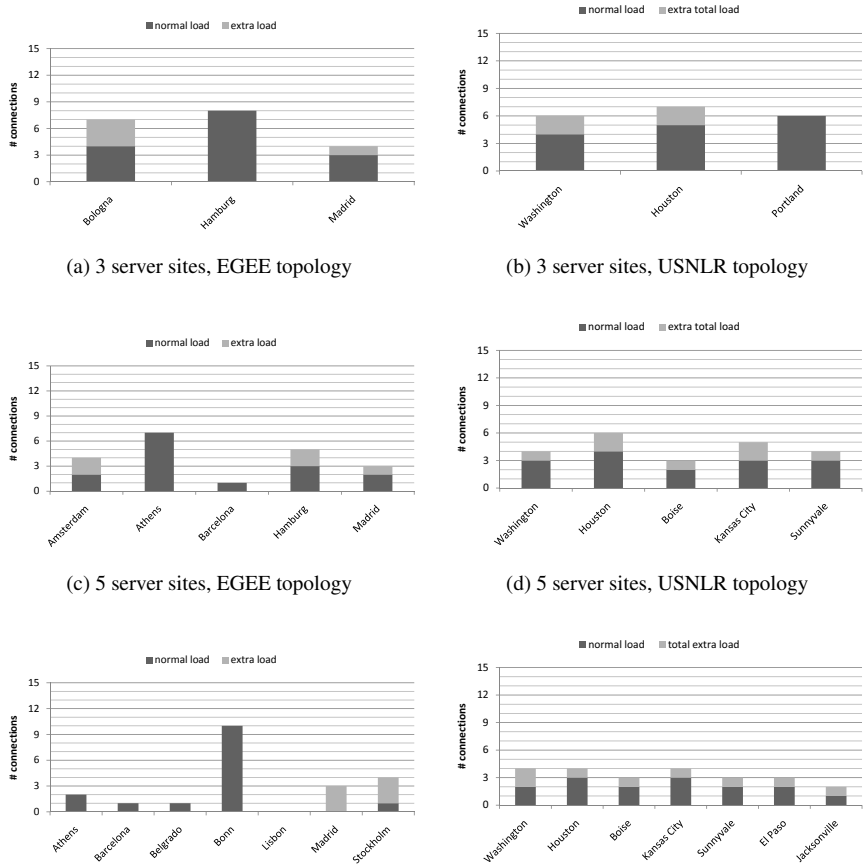
For N_3^{15} (Fig. A.8a) we see that every server site has a failure free load and an extra load. For Bologna, Hamburg and Madrid this extra load is respectively $3/4$, 0 and $1/3$ times its failure free load. Actually 1 connection is only $1/15$ of the total load and if we would express each extra load relative to the load over all servers we end up that every server only caters for respectively $20\%, 0\%$ and 7% of the total load.

Looking at the N_5^{15} case (Fig. A.8c), we see that the load gets more evenly distributed over the different server sites, as is also the case with the extra relocation load.

The last case is N_7^{15} (Fig. A.8e. We notice that not every server receives a failure free load which can be attributed by the mesh property of the network and the small number of source nodes of the network: adding an extra server site to the topology, e.g. going from a N_x^y to a N_{x+1}^y, does not affect the already established clusters of the N_x^y topology. Adding an extra cluster does not mean that a large enough portion of the source nodes is now closer to that extra server site. As a consequence, in the step where the server capacities are chosen (Section A.3.2), the extra cluster does not have a large enough cluster arrival rate and hence, the installed server capacity will be negligible compared to the installed server capacities of the other cluster. Consequently, the scheduling step where the *mostfree* algorithm is used, will schedule only a small part of the jobs to this extra server site. Since there is only a small part of the total load which is sent to this server site, it is rounded down to 0 in the rounding procedure creating the static demand matrix. This is also the reason why Bonn receives a large failure free load: it is the site where the most capacity is installed. However we do see that a server site can be used as dedicated relocation server site (cfr. Madrid) which only receives load in a link-failure scenario.

Focusing on the server site loads for the USNLR case (Fig. A.8b, Fig. A.8d and Fig. A.8f), we see they are somewhat different in nature compared to the EGEE case. For all three cases, each server site receives jobs. The discussion above (for EGEE) does no longer apply for this much sparser (ring-like) USNLR topology.

It is clear that adding an extra server site, e.g. going from a N_x^y to a N_{x+1}^y, is far more profitable in this sparse network case and attracts a reasonably large arrival rate. Therefore the scheduling and rounding steps of the iterative algorithm in A.3 do not result zero unit connection demands towards these added server sites. Accordingly, the notion of an exclusive relocation site disappears. Also, every resource site receives almost an equal part of the relocated jobs (except the N_3^{15} case where Portland does not receive this extra load). Every extra load is either 1 or 2 extra connections which caters for only 6% and 13% of the total requested connections between source and destination sites.



(a) 3 server sites, EGEE topology

(b) 3 server sites, USNLR topology

(c) 5 server sites, EGEE topology

(d) 5 server sites, USNLR topology

(e) 7 server sites, EGEE topology

(f) 7 server sites, USNLR topology

Figure A.7: The server site load in terms of number of arriving connections comprises: (i) in black the number of connections it receives in the failure free case, (ii) in the gray the maximum number of extra connections due to a single link fault. Considering on the one hand the EGEE topology cases, when putting three servers sites into service (Fig. A.8a), we note that each server receives about 10% *of the total load as extra load. The case introducing 7 server locations (Fig. A.8a) exhibits a dedicated relocation server: this server is solely used to cope with relocated jobs. For this EGEE topology, increasing the server site count for the considered demands does not bring added value (in terms of reduced network capacity). For the USNLR topologies on the other hand, we see that increasing the server site count, levels the failure free demand per server. The extra load induced by relocation averages to* 8% *per server and never exceeds* 13% *of the total requested connections. Also, we note that there are no servers exclusively used for relocation.*

A.6 Conclusion

In this work we have described an alternative method for path protection against single link failures in an optical grid scenario. Whereas traditional protection schemes try to reserve backup capacity to the original destination of the primary path, we have accounted for the grid-specific anycast principle (stating that there are several destinations possible for a job to be executed). Therefore, in case of a network failure, we allow to relocate the job to an alternative server site, and as such are able to reduce the bandwidth (wavelengths) to be allocated for the backup path. We have described ILP models for both the traditional shared protection scheme, as well as shared protection with relocation. Our case study pointed out that on average we can achieve a reduction of the total number of necessary wavelengths (network load reduction, NLR) in the range of 17% to 50%, depending on the amount of server sites that have been chosen and the network topology (with a higher NLR for a sparser topology). A sparse network can benefit more of relocation due to the fact that it is more likely to encounter another server on the backup path found in the CSP case.

The NLR is caused by the reduction of backup wavelengths, rather than primary wavelengths. However, the relocation strategy requires adjusted capacities of the relocation servers, since they have to be able to handle these relocated jobs. The amount of extra load is dependent on the number of server sites which have been chosen and again the topology structure. On the one hand, for a meshed European network, we perceived that when selecting 3 server sites we need to provide up to about 20% of the total load as extra capacity). When we increased the number of server sites to 5, this extra load decreased a littel to 7% Increasing the number of server sites is not beneficial for the network dimensions, nor the server resource utilization.

On the other hand, for a sparser US network case study increasing the server site count rather evenly distributes the (failure free) load over the various server sites, as well the extra relocation load. This extra server load now amounts to between 6% and 13%.

References

[1] G. Allen, G. Daues, J. Novotny, and J. Shalf. *The astrophysics simulation collaboratory portal: a science portal enabling community software development*. In Proc. 10th IEEE Int. Symp. High Perf. Distr. Comp. (HPDC), page 207, Washington, DC, USA, 7–9 Aug. 2001. IEEE Computer Society.

[2] B. Allcock, I. Foster, V. Nefedova, A. Chervenak, E. Deelman, C. Kesselman, J. Lee, A. Sim, A. Shoshani, B. Drach, and D. Williams. *High-performance*

remote access to climate simulation data: a challenge problem for data grid technologies. In Proc. ACM/IEEE Conf. Supercomputing (SC), pages 46–46, Denver, CO, USA, 10–16 Nov. 2001.

[3] S. Barnard, R. Biswas, S. Saini, R. Van der Wijngaart, M. Yarrow, L. Zechtzer, I. Foster, and O. Larsson. *Large-scale distributed computational fluid dynamics on the information power grid using globus*. In Proc. 7th Symp. Frontiers of Massively Parallel Comp. (FRONTIERS), page 60, Washington, DC, USA, 21–25 Feb. 1999.

[4] M. De Leenheer, C. Develder, T. Stevens, B. Dhoedt, M. Pickavet, and P. Demeester. *Design and control of optical Grid networks (Invited)*. In Proc. 4th Int. Conf. on Broadband Networks (Broadnets), pages 107–115, Raleigh, NC, 10–14 Sep. 2007.

[5] D. Simeonidou, R. Nejabati, G. Zervas, D. Klonidis, A. Tzanakaki, and M. J. O'Mahony. *Dynamic optical-network architectures and technologies for existing and emerging grid services*. IEEE J. Lightwave Technol., 23(10):3347–3357, Oct. 2005.

[6] D. Colle, S. De Maesschalck, C. Develder, P. Van Heuven, A. Groebbens, J. Cheyns, I. Lievens, M. Pickavet, P. Lagasse, and P. Demeester. *Data-centric optical networks and their survivability*. IEEE J. Sel. Areas Commun., 20(1):6–20, Jan. 2002.

[7] R. Medeiros, W. Cirne, F. Brasileiro, and J. Sauvé. *Faults in Grids: Why are they so bad and what can be done about it?* In Proc. 4th Int. Workshop on Grid Computing (GRID), pages 1–18, Washington, DC, USA, 17 Nov. 2003.

[8] M. Sivakumar, C. Maciocco, M. Mishra, and K. M. Sivalingam. *A hybrid protection-restoration mechanism for enhancing dual-failure restorability in optical mesh-restorable networks*. In Proc. Optical Netw. and Commun. (OptiComm), pages 37–48, Dallas, TX, USA, 13–17 Oct. 2003.

[9] S. Agarwal, R. Garg, M. S. Gupta, and J. E. Moreira. *Adaptive incremental checkpointing for massively parallel systems*. In Proc. 18th ACM Int. Conf. Supercomputing (ICS), pages 277–286, Malo, France, Jun. 26 – Jul. 1 2004.

[10] J. W. Young. *A first order approximation to the optimum checkpoint interval*. ACM Commun., 17(9):530–531, 1974.

[11] Y. Li and M. Mascagni. *Improving performance via computational replication on a large-scale computational grid*. In Proc. 3rd IEEE/ACM Int. Symp. Cluster Computing and the Grid (CCGrid), pages 442–448, Washington, DC, USA, 12–15 May 2003.

[12] C.-J. Hou and K. G. Shin. *Replication and allocation of task modules in distributed real-time systems*. In Proc. 24th Int. Symp. on Fault-Tolerant Comp. (FTCS-24), pages 26–35, Austin, Texas, 15–17 Jun. 1994.

[13] M. Chtepen, F. H. A. Claeys, B. Dhoedt, F. De Turck, P. Demeester, and P. A. Vanrolleghem. *Adaptive task checkpointing and replication: toward efficient fault-tolerant grids*. IEEE Trans. on Parallel and Dist. Systems, 20(2):180–190, Feb. 2009.

[14] G. Maier, A. Pattavina, S. De Patre, and M. Martinelli. *Optical network survivability: protection techniques in the WDM layer*. Photonic Netw. Commun., 4(3–4):251–269, Jul. 2002.

[15] N. Wauters and P. Demeester. *Design of the optical path layer in multiwavelength cross-connected networks*. IEEE J. Sel. Areas Commun., 14(5):881–892, Jun. 1996.

[16] M. Tornatore, G. Maier, and A. Pattavina. *WDM network design by ILP models based on flow aggregation*. IEEE/ACM Trans. Netw., 15(3):709–720, 2007.

[17] J. Buysse, M. De Leenheer, C. Develder, and B. Dhoedt. *Exploiting relocation to reduce network dimensions of resilient optical grids*. In Proc. 7th Int. Workshop Design of Reliable Commun. Netw. (DRCN), pages 100–106, Washington, D.C., USA, 25–28 Oct. 2009.

[18] H. Zang, C. Ou, and B. Mukherjee. *Path-protection routing and wavelength assignment RWA in WDM mesh networks under duct-layer constraints*. IEEE/ACM Trans. Netw., 11(2):248–258, 2003.

[19] C. Develder, B. Mukherjee, B. Dhoedt, and P. Demeester. *On dimensioning optical grids and the impact of scheduling*. Photonic Netw. Commun., 17(3):255–265, Jun. 2009.

[20] K. Christodoulopoulos, E. Varvarigos, C. Develder, M. De Leenheer, and B. Dhoedt. *Job demand models for optical grid research*. In Proc. 11th Int. IFIP TC6 Conf. on Optical Netw. Design and Modeling (ONDM), pages 127–136, Athens, Greece, 29–31 May 2007.

[21] B. Van Houdt, C. Develder, J. F. Pérez, M. Pickavet, and B. Dhoedt. *Mean field calculation for optical grid dimensioning*. IEEE/OSA J. Opt. Commun. Netw., 2(6):355–367, Jun. 2010.

B

Calculating the minimum bounds of energy consumption for cloud networks

Buysse, J.; Georgakilas, K.; Tzanakaki, A.; De Leenheer, M.; Dhoedt, B.; Demeester, P. & Develder, C., *Calculating the minimum bounds of energy consumption for cloud networks*, published in Proceedings of the IEEE International Conference on Computer Communications and Networks (ICCCN 2011), Maui, Hawai, US, pp. 1-7, July 31 - August 4 2011

This paper has been included in the dissertation, as it forms the base for the work described in Chapter 5.

Abstract This paper is aiming at facilitating the energy-efficient operation of an integrated optical network and IT infrastructure. In this context we propose an energy-efficient routing algorithm for provisioning of IT services that originate from specific source sites and which need to be executed by suitable IT resources (e.g. data centers). The routing approach followed is anycast, since the requirement for the IT services is the delivery of results, while the exact location of the execution of the job can be freely chosen. In this scenario, energy efficiency is achieved by identifying the least energy consuming IT and network resources required to support the services, enabling the switching off of any unused network and IT resources. Our results show significant energy savings that can reach up

to 55% compared to energy-unaware schemes, depending on the granularity with which a data center is able to switch on/off servers.

B.1 Introduction

It is widely accepted that Internet traffic is growing very fast [1]. As such, the energy consumption of the ICT becomes significant and cannot be neglected any more. Several studies have pointed out that ICT around the world is responsible for up to 10% of the total energy consumption and 2% of global carbon emissions [2].

In the context of Future Internet and cloud computing, integration of IT and network resources in a common infrastructure that supports a large variety of existing and future services also becomes a necessity. Cloud computing entails a system to access a set of computing resources such as computational, data and software services in an on-demand and convenient way, without the end-user interacting with the hardware or service provider [3]. Consequently the network supporting the cloud should be able to bear large data transfers in a fast and reliable way. Given their high data rates and low latency, optical networks based on wavelength division multiplexing (WDM) technology are ideally suited. These considerations motivate us to focus our attention on reducing the energy consumption of integrated optical network and IT infrastructures. It is clear that in order to identify an optimal solution achieving minimum energy consumption, joint consideration of both network and IT resources will be required.

Examining this type of infrastructure, one can identify the following elements to consider from an energy consumption perspective: optical links, optical switching nodes and data centers. In this work we aim at reducing the overall energy consumption of such an infrastructure by employing two strategies:

1. Switching off components when they are in an idle state.

2. Exploiting the anycast principle to provision IT requests to the most appropriate data center and compute routes in an energy-efficient way.

Anycast is based on the principle that a user is not concerned with the exact location of the execution of the submitted IT request, as long as the requirements of the service are met. Hence, when operating an infrastructure such as the one described above, the selection of both the destination IT site and the network resources that allow the routing of the IT service from a remote user, can be based on the associated energy consumption. This can be performed by including the relevant energy parameters in the objective of the associated optimization, as opposed to unicast which is less flexible because the destination IT site is known a priori.

The remainder of the paper is organized as follows. In Section B.2 we introduce related work focusing on energy efficiency in optical networks and, in Section

Section B.3 we formulate the power models for the optical network and the IT resources. In Section Section B.4 we formally articulate the problem and provide a Mixed Integer Linear Programming (MILP) approach. In Section Section B.5 we provide a use case scenario providing results and insights. Finally we conclude the paper in Section Section B.6.

B.2 Related Work

The work in [4] reports a detailed study to estimate the impact of ICT on the environment in general and on energy needs in particular. According to predictions made, it is clear that the pressure on power efficiency in ICT will become more and more prominent in the coming years and needs to be dealt with accordingly.

The authors in [5] have investigated the influence of the availability of switching off network elements under connectivity and Quality of Service (QoS) constraints. Results show that it is possible to reduce the number of active links and nodes up to 25%. This work confirms that there is a network energy optimization to be made by switching off resources. We extend the principle further by also considering IT power consumption and exploiting the anycast principle.

The work described in [6] provides a comprehensive survey of the most relevant research activities for minimizing energy consumption in telecom networks, with specific emphasis on those employing optical technologies. Energy minimization opportunities enabled by optical technologies are investigated and classified over different network domains, namely core, metro and access networks.

An investigation of the potential savings achievable through power-aware network design and routing is presented in [7]. The authors have conducted measurements of the power consumption in various configurations of widely used core and edge routers and have explored the potential impact of power-awareness in a set of example networks. Results indicate that power consumption can vary by as much as an order of magnitude, indicating that there may be substantial opportunities for reducing power consumption.

An analysis of several designs for green routing algorithms is presented in [8]. The authors formulate the problem as a minimum energy routing optimization, where nodes cannot be switched off (as opposed to our work). It is demonstrated that depending on the topology and traffic matrices, the optimal energy savings can be modest for some scenarios. The authors also counteract the belief that there exists a trade-off between energy-efficient network optimization and performance.

The work presented in this paper extends previous work in two ways. We first develop a generic energy model for the integrated IT and optical network infrastructure where energy is consumed by the optical switching nodes and links as well as the data centers so as to treat network and IT power as part of the whole optimization objective. Secondly we are considering the anycast principle,

as opposed to other works where a static traffic matrix is assumed. This allows us to choose the destination sites, in order to decrease the overall power consumption. Thirdly, we benchmark the proposed strategy compared to other traditional routing and allocation schemes in an extensive case study.

B.3 Power consumption models

We aim to minimize the energy used by both the data centers and the optical network. Therefore it is imperative to rely on models that accurately describe the power consumption of the associated devices. In this work we assume optical switching nodes that are regenerating, wavelength convertible optical cross-connects based on a central optical switching fabric using 3D MEMS (Micro-Electro-Mechanical Systems) switch technology [9] described in detail in Section Section B.3.2. For the data center power consumption model we deploy a flexible power estimation framework described in Section B.3.1.3.

B.3.1 IT power model

B.3.1.1 Computer power consumption index

As a data center can house hundreds or even thousands of servers and storage devices it is obvious that it is quite energy intensive. Apart from the servers, there are several other factors that add to the power consumption such as backup generators, switching gear, cooling systems, uninterruptible power supplies (UPS) and Power Distribution Units (PDU). To express this extra power consumption for a data center, we use the *computer power consumption index* (the inverse of Power Usage Effectiveness, PUE) which is the fraction of the total energy consumption of the data center to that used by the servers housed in the data center. The authors of [10] have investigated this index for 22 data centers and concluded that this ranges from a very poor 0.33 up to 0.75 where on average it is about 0.5. This means that half of the energy consumption of a data center goes to cooling etc. For OXCs, there is a similar index, which is of the same kind (about 0.5). Therefore, in the rest of the paper, we omit this extra power as it does not modify the relative network vs. IT power.

B.3.1.2 Power Consumption of a server

As the authors of [11] have demonstrated, linear Eq. B.1 produces quite accurate estimations for a server's power consumption, given the load (α), the power in idle state (P_{idle}) and the power when at 100% load (P_{max}). In this work, we express load in flops (FLoating point OPerations per Second). We use the same metric

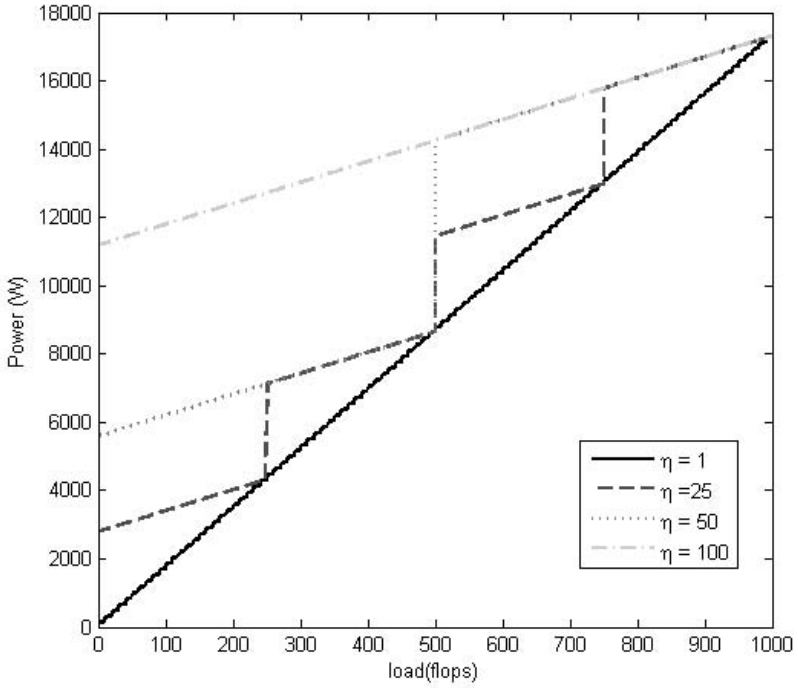


Figure B.1: Power consumption of a data center with $\eta \in 1, 25, 50, 100$ with 100 servers with server power characteristics $P_{max} = 118W$ and $P_{idle} = 56.7W$.

when expressing the capacity of a server (z) i.e. the maximum number of flops it is able to serve.

$$P_{cpu}(\alpha) = P_{idle} + \frac{(P_{max} - P_{idle})}{z} \cdot \alpha \tag{B.1}$$

B.3.1.3 Power consumption of a data center

For this study, we assume a data center which is able to switch off all servers in a certain rack when all servers in that rack are idle. To express the granularity in which a data center can switch off servers, we introduce the parameter η, which corresponds to a number of servers in a rack. For example when $\eta = 1$ the data center can switch on/off each single server, whereas when $\eta = 5$ the data center has to switch on/off a rack with five servers. Note that these servers consume their idle power P_{idle} when turned on. As shown in Section B.5 this parameter directly influences the routing and IT request allocation scheme.

Accordingly, the power consumption of a data center is expressed in Eq. B.2 (α = total load offered to the data center). In Fig. B.1 we have plotted the power consumption of a data center (a stepwise function), with different values for η, depending on the load.

$$P_{DC}(\alpha) = \left\lfloor \frac{\alpha}{z \cdot \eta} \right\rfloor \cdot P_{max} + \eta \cdot P_{idle} + \frac{P_{max} - P_{idle}}{z} \cdot \left(\alpha - \left\lfloor \frac{\alpha}{z \cdot \eta} \right\rfloor \right) \quad \text{(B.2)}$$

B.3.2 Network power model

The node architecture considered in this work is an optical cross-connect based on an optical switching fabric. The OXC supports N input and N output fibers, each employing a maximum number of wavelengths, W. The total number of ports is the sum of express (through) and add/drop ports. To overcome the limitations that the wavelength continuity constraint imposes in optical networks, we assume full wavelength conversion capability. This is facilitated through the allocation of a wavelength converter at the output of every through switching port based on conventional optoelectronic transponder technology offering at the same time signal regeneration. Moreover, the OXC architecture employed supports the ability to add/drop up to 50% of the total through traffic. As shown in Fig. B.2, one transmitter for each add port and one receiver for each drop port is needed. The total power consumption of the node depends on four parts: the switch fabric, the wavelength converters (transponders), the transmission equipment (transmitters (add),receivers (drop)), and the optical amplifiers based on Erbium Doped Fiber Amplifier (EDFA) technology.

Hence, the power consumption of the OXC n is then calculated as follows (all parameters are explained in Section B.4):

$$P_n = P_{mems} + P_{oeo} + P_{ampl} + P_{trans} \quad \text{(B.3a)}$$

$$P_{mems} = \rho_{total} \cdot P_{pair} \quad \text{(B.3b)}$$

$$P_{oeo} = \rho_{through} \cdot P_p \quad \text{(B.3c)}$$

$$P_{ampl} = (\omega^+(n) + \omega^-(n)) \cdot P_{edfa} \quad \text{(B.3d)}$$

$$P_{trans} = \rho_{a/d} \cdot P_{Tx/Rx} \quad \text{(B.3e)}$$

In equation Eq. B.3 ρ_{total} represents the total number of switch ports and $\rho_{through}$ the the number of express (through) ports. The total power consumption of the optical network can be derived by the addition of the power consumption of all active switching nodes and links as described in detail in [12] and therefore it can be expressed as indicated by Eq. B.4 including the energy consumption of both optical switching nodes and links. The power consumption of optical links is

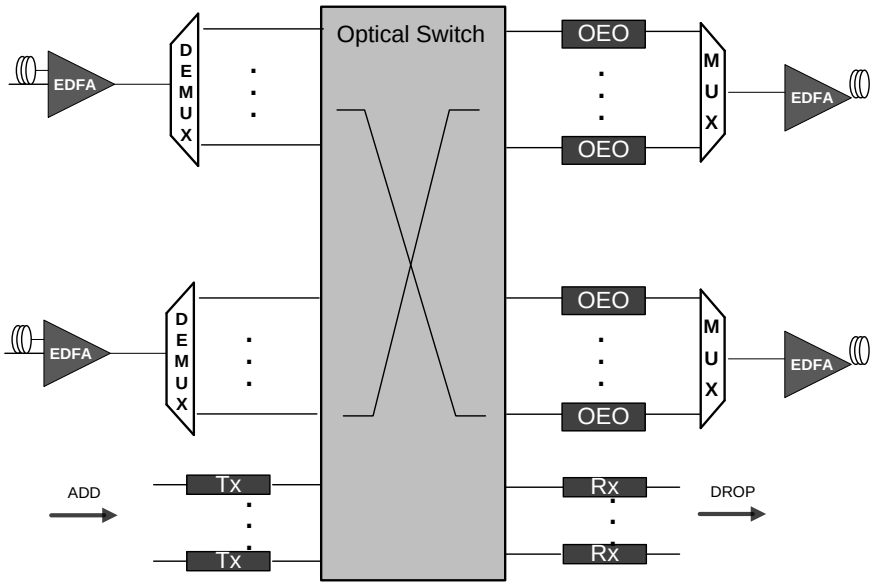


Figure B.2: The OXC architecture, illustrating the power-dissipating elements of the OXC with gray color.

attributed to the optical amplifiers used to compensate the insertion loss associated with signal transmission over optical fibers. The distance between consecutive amplifiers is referred to as the *fibre span*.

$$P_{netw} = \sum_{n \in V} P_n + \sum_{l \in E} \left\lceil \frac{|l|}{\textit{fibre span}} \right\rceil \cdot P_{\textit{edfa}} \tag{B.4}$$

B.4 Formal problem statement and formulation

As opposed to traditional routing schemes, we are not assuming a traffic matrix representing the number of connections requested between each source and destination site. For the offline provisioning of the network we assume a traffic vector expressing a number of desired connections between a source and some server site in the network that is not predetermined. Hence, it is up to the MILP to decide which server site best suites the objective function. Furthermore we assume that every OXC is able to perform full wavelength conversion.

For the IT part, we assign a number of flops each request expresses its need for. For simplicity we assume the same number of flops per request.

The MILP employs the following parameters:

$G = (N, L)$, a graph representing an optical network

N Node set, indexed by $n \in N$

L The link set, with 1 fibre per link, index by l

K Request set, indexed by $k \in K$

k_n Request originating at n

z_n $\in [0, \infty[$. The maximum number of flops server n can process

m number of servers in a data center

η switch on/off granularity of a data center

f^k $\in [0, \infty[$. The number of flops request k needs

P_{mems} power consumption of the switching fabric (e.g. MEMS),

P_{oeo} power consumption attributed for wavelength conversion (e.g. by using the OEO converters)

P_{ampl} power consumption of all the amplifiers of the OXC

P_{trans} power consumption required by the transmitters and receivers

P_{pair} power consumption per input/output port pair [13], 0.107 W

P_{edfa} power consumption of an EDFA, 13 W

P_c power consumption of a receiver, 3.5 W

P_m power consumption of a transmitter, 3.5 W

P_p power consumption of a transponder (OEO converter), 6 W

P_{max} power consumption of a server at full load 118 W

P_{min} power consumption of a server in idle state 56.7 W

e use the following notations:

$\omega^-(n)$ The outgoing fibers of OXC n

$\omega^+(n)$ The incoming fibers of OXC n

$\omega(n)$ $\omega^-(n) \cup \omega^+(n)$

$|l|$ length of link l

The variables in the MILP are as follows:

$w_l \in [0, \infty[$ number of active wavelengths on $l \in L$. We will assume all links have the same maximum transport capacity (in terms of wavelengths), say $w_l \leq W$ for all $l \in L$

$w_l^k \in \{0, 1\}$ Is 1 if request k is routed over fiber l, 0 otherwise

$f_l \in \{0, 1\}$ Is 1 if fiber l is used, meaning at least one of its wavelengths is activated

$y_n \in \{0, 1\}$ Is 1 if OXC n is powered on, 0 otherwise

$x_n \in \{0, 1\}$ Is 1 if data center n is powered on, 0 otherwise

$b_n^k \in \{0, 1\}$ Is 1 if data center n processes request k

$b_n \in [0, \infty[$ The total number of requests a data center n is processing

$\alpha_n \in [0, \infty[$ Load offered to data center n, in flops.

$\beta_n \in [0, \infty[$ Number of racks which are switched on, in data center n

$\gamma_n \in [0, \infty[$ The power used by all servers at full load in data center η

$d_n \in [0, \infty[$ The power consumption of data center n

$o_n \in [0, \infty[$ The power consumption of OXC n

$s_n \in [0, \infty[$ The number of switched paths in node n

$t_n \in [0, \infty[$ The number of paths terminated in node n

$r_n \in [0, \infty[$ The number of locally processed jobs in node n

B.4.1 Objectives

We have implemented four different objectives. The first objective Eq. B.5 minimizes only the network energy (referred to as N), the second objective Eq. B.6 minimizes the IT energy and applies shortest path routing (referred to as I), the third objective Eq. B.7 applies shortest path routing while the last objective Eq. B.8 minimizes both IT and network energy (referred to as NI). We have used a δ variable in order to achieve shortest path routing in I, as otherwise random routes would be taken. In order to achieve this shortest path routing in I, we need to keep this δ very small therefore we used 0.001.

Objective N

$$\min \left(\sum_{n \in N} o_n + \sum_{l \in L} f_l \cdot \left\lceil \frac{l}{\textit{fibre span}} \right\rceil \cdot P_{\textit{edfa}} \right) \tag{B.5}$$

Objective I

$$\min\left(\sum_{n\in N} d_n + \delta \cdot \sum_{l\in L} w_l \cdot |l|\right) \tag{B.6}$$

Objective SP

$$\min\left(\sum_{l\in L} w_l \cdot |l|\right) \tag{B.7}$$

Objective NI

$$\min\left(N + I\right) \tag{B.8}$$

B.4.2 Constraints

B.4.2.1 Network Modeling

We start by formulating the flow conservations:

$$\sum_{l\in\omega^+(n)} w_l^k - \sum_{l\in\omega^-(n)} w_l^k = \begin{cases} -1 & \text{if } n \text{ is } k\text{'s source} \\ b_n^k & \text{otherwise} \end{cases} \quad n \in N, k \in K. \tag{B.9}$$

The next set of constraints represent the demand constraints:

$$\sum_{n\in N} b_n^k = 1 \qquad \forall k \in K \tag{B.10a}$$

$$b_n = \sum_{k\in K} b_n^k \qquad \forall n \in N \tag{B.10b}$$

$$b_n \leq z_n \qquad \forall n \in N \tag{B.10c}$$

We enforce the network capacity constraints:

$$w_l = \sum_{k\in K} w_l^k \qquad \forall l \in L \tag{B.11a}$$

$$w_l \leq W \qquad \forall l \in L \tag{B.11b}$$

In the next set of constraints we calculate whether a link or an OXC is turned on or not. (M is the node degree of n):

$$f_l \leq w_l \qquad \forall l \in L \tag{B.12a}$$

$$f_l \cdot W \geq w_l \qquad \forall l \in L \tag{B.12b}$$

$$\sum_{l\in\omega(n)} f_l \leq M \cdot y_n \qquad \forall n \in N \tag{B.12c}$$

$$y_n \leq \sum_{l\in\omega_n} f_l \qquad \forall n \in N \tag{B.12d}$$

We apply these constraints to compute the number of dropped, added and switched paths:

$$r_n = \sum_{K_n \in K} b_n^k \qquad \forall n \in N \tag{B.13a}$$

$$s_n = \left(\sum_{l \in \omega^+(v)} w_l \right) - t_n \qquad \forall n \in N \tag{B.13b}$$

$$t_n = b_n - r_n \qquad \forall n \in N \tag{B.13c}$$

This last constraint represents the power consumption of the OXC, as described in Section B.3.2.

$$o_n = \sum_{l \in \omega(n)} f_l \cdot P_{edfa} + y_n \cdot P_{mems} + t_n \cdot P_c + (K_n - r_n) \cdot P_m + s_n \cdot P_p \tag{B.14a}$$

$\forall n \in N$

B.4.2.2 IT modeling

We apply demand constraints for the IT requests.

$$\sum_{k \in K} b_n^k \cdot f^k \leq z_n \cdot m \forall n \in N \tag{B.15}$$

Next, we check whether a data center is turned on only if it processes requests.

$$x_n \cdot M \geq b_n \qquad \forall n \in N \tag{B.16a}$$

$$x_n \leq b_n \qquad \forall n \in N \tag{B.16b}$$

The next set of constraints are used to compute the power of a data center, depending on η ($\beta \in \mathbb{N}$).

$$\alpha_n = \sum_k b_n^k \cdot f^k \qquad \forall n \in N \tag{B.17a}$$

$$\gamma_n = \beta_n \cdot z_n \cdot \eta \qquad \forall n \in N \tag{B.17b}$$

$$\gamma_n \leq \alpha \qquad \forall n \in N \tag{B.17c}$$

$$\gamma_n \geq \alpha - (z_n \cdot \eta + 1) \qquad \forall n \in N \tag{B.17d}$$

Figure B.3: The reference topology [14] used for obtaining the results.

And finally we compute the power of a data center (only if $z_n > 0$) according to Section B.3.1.3.

$$d_n = \beta_n \cdot \eta \cdot P_{max} + \eta \cdot P_{idle} + \frac{(P_{max} - P_{idle})}{z_n} \cdot (\alpha - \gamma_n) \quad \forall n \in N \quad \text{(B.18a)}$$

B.4.3 Complexity

The scalability and complexity of a MILP mainly depends on the number of variables and constraints which are employed. For this MILP the number of variables is $|N| \cdot (11 + |K|) + |L| \cdot (2 + |K|)$ and the number of constraints is $16 \cdot |N| + |K| \cdot (|N| + 1) + 4 \cdot |L|$. As can be observed the MILP is scaling with the number of requests and the number of nodes in the network.

B.5 Use Case

The network topology [14] used in this work is the European topology (shown in Fig. B.3), which is a result of a joint effort from the IST LION project and the COST 266 action project [15]. We assumed 5 server sites, each with 20 servers (ASUS RS160-E5 Intel Xeon L5420 Processor, 2.50 GHz [16]), located at Berlin, London, Lyon, Vienna and Zurich. For the network we assumed 20 wavelengths per fiber. We generated 10 random demand vectors per demand instance ranging from 10 to 100 requests where each request needed one server ($f^k = z_n$ and 1 wavelength path towards the respective server site). Consequently, the results shown in the graphs represent averages of these 10 demand instances. Note that, as there are five server sites each incorporating 20 servers, 100 IT requests correspond to a load that requires all datacenters to be working at full capacity.

B.5.1 Network energy aware routing vs. Network+IT energy aware routing

In Fig. B.4 we show the distribution of the different power dissipating elements and the extent to which they take part in the total energy consumption for the scenario with $\eta = 1$. In this scheme there are no means to optimize the IT resource allocation, as every IT request can be scheduled to a server without the need to switch on other unnecessary servers. Hence, this is the minimum IT power needed to accommodate the IT load. We notice that, independent of the load, predominant energy consuming resources are the data centers (with their corresponding servers). This result indicates that intelligent IT resource allocation will probably be more beneficial than energy aware allocation of network resources only. This is demonstrated in Fig. B.5 where the parameter $1 - \frac{P^N_{netw}}{P^{NI}_{netw}}$ has been plotted. P^X_{netw} is the total network power, for the MILP with objective minimization X. This data represents the extra percentage of network power needed to accommodate for the optimal energy aware IT resource allocation NI, compared to the pure network energy minimization objective N, in order to achieve the reduction of the total energy shown in Fig. B.6 (up to 55% for $\eta = 20$). There are two observations in Fig. B.5: by allowing a suboptimal solution for the network routing we (i) enable a general decrease in the overall power consumption due to improved scheduling of IT requests and (ii) for increasing η this extra fraction of network power generally increases (together with the difference in power use). This can be explained by the fact that for bigger η, the use of a server always introduces powering on a whole rack of η servers. Given that IT power consumption is dominant (see Fig. B.4) one strives to fully utilize complete racks in the NI optimization, leading to longer (more network power consumption) paths than strictly necessary.

Concluding it is clear that when the intention is to minimize the total energy

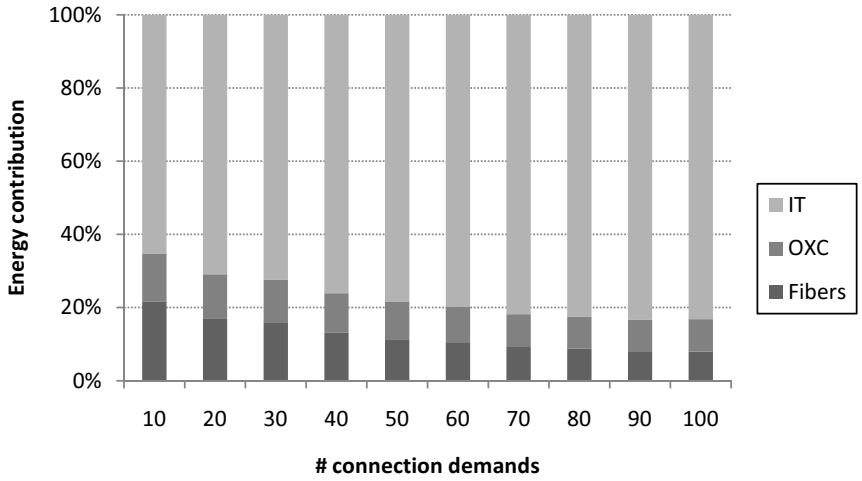


Figure B.4: Relative power consumption of OXCs, links and data centers compared to the total power consumption for $\eta = 1$.

consumption in an integrated network and IT scenario, the optimal solution should be a careful consideration of combined network and IT resource energy parameters.

B.5.2 Comparing the routing schemes with different objectives

As indicated in Section B.4.1 we have opted for four different objectives: (i) shortest path routing (SP), (ii) minimization of network energy (N), (iii) minimization of IT energy with SP routing (I) and (iv) minimization of network and IT energy (NI). In Fig. B.6 we have shown the total energy consumption values for the different objectives, for different η.

We first focus on the pure network objectives N and SP where we notice a similar power consumption. Both result in a similar total power with a difference of ca. 2% and from a total energy perspective the best objective is N. This observation can be attributed to the fact that the N scheme in most cases selects the closest server site (the same as SP) and therefore both SP and N objectives reach the same IT power. So the only difference in the energy consumption can be attributed to intelligently routing paths in order to allow switching off certain links and nodes and increase the sharing of network resources among paths. This way we can achieve a *network* energy reduction up to 10% compared to SP (depending on the load), but due to the balance between IT and network power (ca. 20% Network power vs. ca. 80% IT power, see Fig. B.4), this decrease is translated into a small percentage of the overall energy consumption of the infrastructure (on average 2%). Moreover, this difference in the total energy consumption between

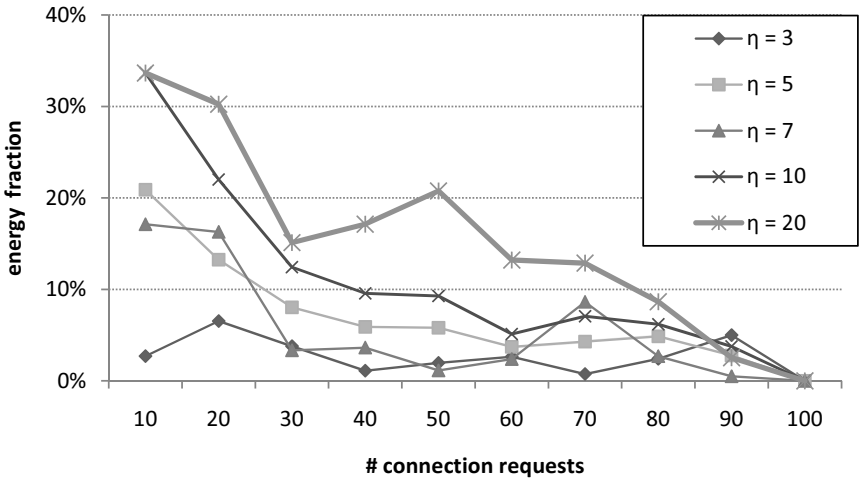


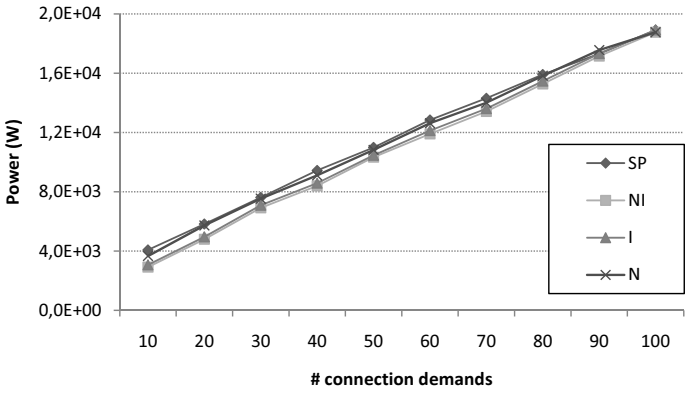
Figure B.5: The percentage of extra power needed to accommodate for intelligent routing in the NI case, compared to the pure network energy optimization case.

Network and SP becomes smaller as η increases, because then the IT resource power consumption becomes even more dominant.

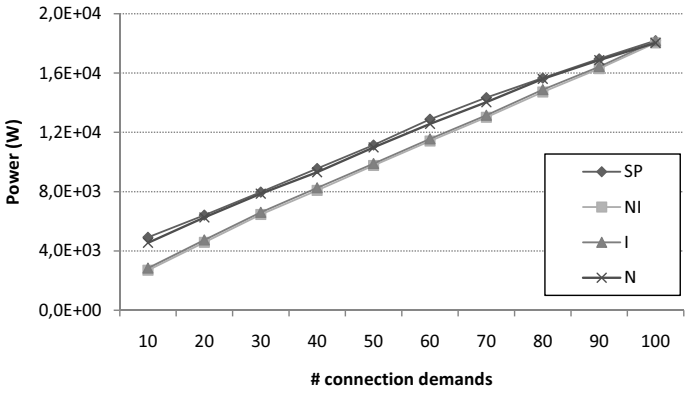
When comparing the IT-aware objectives I and NI, we once more note that there is little or no difference (ca. 2%) for the total energy consumption, with NI always providing the optimal solution. When we differentiate between IT and network energy, we note that there is never a difference between objectives regarding IT power consumption (as was suggested in Section B.5.1). NI can decrease its network power consumption by 5% to 10% compared to IT-only minimization, by providing energy-efficient routes the same way as already indicated above. However, as this network power decrease only accounts for a small portion of the total energy consumption (see Fig. B.4), this decrease is hardly noticeable (a decrease of 2%). Observing the unused and still available network resources it is seen that there is little or no difference between the two objectives : the free capacity in NI is only 1%-2% lower than that for I. These results clearly indicate that there is practically no penalty in the efficiency of the resource utilization introduced through the reduction of the network power consumption.

In Fig. B.6 we show the total power consumption graphs for $\eta \in [7, 10, 20]$. We see that with increasing η, the difference between the pure network objectives (SP and N) and the objectives incorporating IT power parameters (I and NI) increases. This can be explained by the fact that the requirement to switch on a rack of servers, when allocating an IT request to an element of that rack, increases the penalty brought on by IT-unaware routing which is too large compared to the potential network savings. We conclude that the selection of the allocation scheme

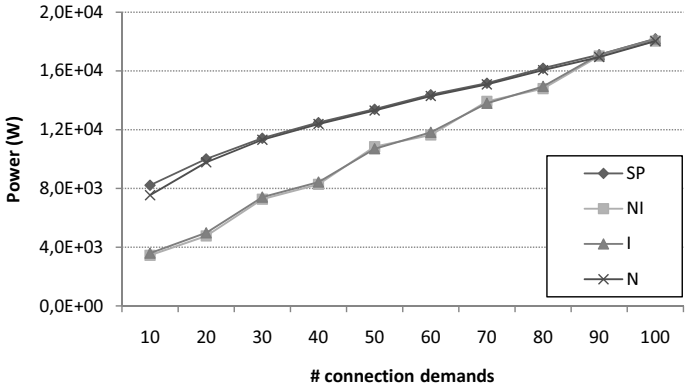
could depend on the granularity of set of servers (e.g. a rack) that can jointly be turned on/off. If η is rather small ($\eta \in [1, 2, 3]$), the only considerable optimization is one for the network power. If the absolute minimum energy consumption is targeted, N is to be favored. If some tolerance is allowed, SP will yield an acceptable solution in a shorter time frame. As η increases ($\eta \geq 4$), the IT-unaware routing introduces a high IT related energy penalty (up to 55% for $\eta = 20$) and the combined resource allocation scheme needs to be considered if the absolute minimum is requested, while I will yield acceptable results if some margin is allowed.



(a) $\eta = 7$



(b) $\eta = 10$



(c) $\eta = 20$

Figure B.6: Total power consumption, for each optimization objective (SP,N,I and NI) for $\eta = 7, 10, 20$.

B.6 Conclusion and future work

Energy considerations in ICT are becoming of significant importance, as it is shown that ICT is responsible for about 10% of the global energy consumption. Therefore this work addresses the energy efficient operation of integrated network and IT infrastructures in the context of cloud computing. By allowing to switch off several IT and network elements and by exploiting the anycast principle, we propose an energy-efficient routing and IT allocation algorithm, using MILP. Results gathered from a use case on an European topology, demonstrated that the predominant energy consuming resources are the servers installed in the data centers, as they are responsible for ca. 80% of the total power consumption. If only the network energy consumption is taken into account in deciding to which IT server site requests are allocated, considerable energy waste may be introduced. More specifically, comparing joint minimization of both network and IT energy provides energy savings of the order of 3%-55% compared to the network energy minimization only approach, depending on the ability of a data center to switch on/off a set of servers (e.g. a rack). On the other hand, pure network-energy minization allows energy savings of the order of 1-2% of the total energy budget compared to shortest path routing (i.e. energy-unaware). Future work includes investigating the effect of choice of server site locations on the energy savings and creating a more scalable method of computation (e.g. column generation, heuristics, etc.).

References

[1] *Usage and population Statistics*. Technical report, Internet World Stats, http://www.internetworldstats.com/stats.htm, 2011.

[2] *An inefficient truth - executive summary*. Technical report, Global Action Plan, www.globalactionplan.org.uk, 2007.

[3] M. Peter and G. Timothy. *The NIST definition of cloud computing*. Technical report, Institute of Standards and Technology, http://csrc.nist.gov/publications/drafts/800-145/Draft-SP-800-145_cloud-definition.pdf, Jan 2011.

[4] M. Pickavet, W. Vereecken, S. Demeyer, P. Audenaert, B. Vermeulen, C. Develder, D. Colle, B. Dhoedt, and P. Demeester. *Worldwide energy needs for ICT- The rise of power-aware networking*. In Proc. of the 2nd Int. Symp. on Adv. Netw. and Telecommun. Systems (ANTS), pages 1–3, 2008.

[5] L. Chiaraviglio, M. Mellia, and M. Neri. *Energy-aware networks: reducing power consumption by switching off network elements*. In FEDERICA-Phosphorus tutorial and workshop (TNC2008), May 2008.

[6] Z. Yi, P. Chowdhury, M. Tornatore, and B. Mukherjee. *Energy efficiency in telecom optical networks*. IEEE Commun. Surveys and Tutorials, 12(4):441–458, 2010.

[7] J. Chabarek, J. Sommers, P. Barford, C. Estan, D. Tsiang, and S. Wright. *Power awareness in network design and routing*. In Proc. of the 27th Conf. on Comp. Commun. (INFOCOM), pages 457–465, Phoenix, AZ, U.S.A., Apr. 2008.

[8] A. Bianzino, C. Chaudet, F. Larroca, D. Rossi, and J. Rougier. *Energy-aware routing: A reality check*. In Proc. of 3rd Int. Workshop on Green Commun. (GreenComm), in conjunction with GLOBECOM, Miami, FL, USA, 6–10 Dec. 2010.

[9] M. Makoto. *Analyzing power consumption in optical cross-connext equipment for future large-capacity optical networks*. J. of Netw., 5(11):1–4, 2010.

[10] S. Greenberg, M. Evan, T. Bill, R. Peter, and M. Bruce. *Best practices for data centers: lessons learned from benchmarking 22 data centers.* Proc. of the ACEEE Summer Study on Energy Efficiency in Buildings in Asilomar, CA. ACEEE, 3:76–87, 2006.

[11] X. Fan, W. Weber, and L. Barroso. *Power provisioning for a warehouse-sized computer*. In Proc. of the 34th Annual Int. Symp. on Computer Arch., SCA, pages 13–23, New York, NY, USA, 09 –11 Jun. 2007. ACM.

[12] A. Tzanakaki, K. Katrinis, T. Politi, A. Stavdas, M. Pickavet, P. Van Daele, D. Simeonidou, M. J. O. Mahony, A. Slaviša, L. Wosinska, and P. Mont. *Power considerations towards a sustainable Pan-European network*. In Proc. of the Nat. Fiber Optic Engineers Conf. (NFOEC/OFC), Mar. 2011.

[13] A. Slaviša. *Analysis of power consumption in future high-capacity network nodes*. J. of Optical Commun. Netw., 1(3):245–258, 2009.

[14] S. De Maesschalck, D. Colle, I. Lievens, M. Pickavet, P. Demeester, C. Mauz, M. Jaeger, R. Inkret, B. Mikac, and J. Derkacz. *Pan-European optical transport networks: An availability-based comparison*. Photonic Netw. Commun., 5(3):203–225, May 2003.

[15] A. Kuchar. *Achievements of COST 266 Action and further prospects in research of advanced infrastructure for photonic networks*. In 6th Int. Conf. on Trans. Optical Netw., 2004., volume 1, pages 37–42 vol.1, Jul 2004.

[16] S. P. E. Corporation. *SPECpower*. Technical report, SPEC, http://www.spec.org/power_ssj2008/, 2008.